Analysis of information quality requirements in business processes, revisited

Noname manuscript No.
(will be inserted by the editor)

Analysis of Information Quality Requirements in Business Processes, Revisited

Mohamad Gharib · Paolo Giorgini · John Mylopoulos

Received: date / Accepted: date

Abstract Many business processes (BPs) involving critical decision-making activities require good quality information for their successful enactment. Despite this fact, existing BP approaches focus on control-flow and ignore the complementary information perspective, or simply treat it as a technical issue, rather than a social and organizational one. To tackle this problem, we propose a comprehensive framework for modeling and analyzing information quality requirements for business processes using the WFA-net BP modeling language. In addition, we describe a prototype implementation, and present two realistic examples concerning the stock market domain, intended to illustrate our approach.

Keywords Information Quality · Business Process · Workflow Nets · Requirements Engineering · Socio-technical Systems

1 Introduction

A Business Process (BP) can be defined as a set of activities with a clear structure describing their sequencing order and dependencies [1]. Although information-related problems often constitute a primary reason for failure [77], existing BP approaches focus on control/activity flow, with less emphasis on information (e.g., [50, 23, 43, 87]). However, some efforts have been devoted to information-aware process design (e.g., Sadiq et al. [64]; Sidorova et al. [68]; Trčka et al. [77]). In particular, these works combine information flow with activity flow, i.e., they are able to detect information unavailability in BPs, but none of them consider Information Quality (IQ) related issues in BPs. IQ is a key success factor for most BPs, since low-quality information may result in undesirable outcomes [57], or it might even prevent the BP from achieving its goals. IQ can be defined as "fitness for use" [40], i.e., meeting or exceeding users' expectations. However, IQ is a hierarchical multi-dimensional concept that can be characterized through different dimensions, e.g., accuracy, completeness, timeliness, etc. [57, 54, 9]. This makes determining whether IQ requirements are satisfied harder, since it has to be done by analyzing several dimensions.

In the literature, we can find several techniques for preventing [56, 55], detecting [15, 70], and correcting [34, 7] IQ-related issues. However, most of these techniques propose solutions that are able to address technical aspects of IQ, while ignoring social and organizational aspects. But such aspects are very important, since BPs are mostly enacted by social actors, rather than machines [22]. More specifically, most BPs these days are executed in a social context (e.g., socio-technical systems [21]), where humans and technical components are both integral part of the BP. Therefore, understanding the social and organizational context where the BP is enacted is essential to detect different kinds of vulnerabilities that might influence BP enactment. Fisher and Kingma [24] showed how existing IQ techniques are not able to handle IQ needs for socio-technical systems, where different kinds of vulnerabilities might manifest themselves in actor interactions and dependencies.

The Flash Crash (a major US stock market crash [72]) is an example where the problem was not caused

M. Gharib
University of Trento - DISI, Povo, Trento, Italy
E-mail: mohamad.gharib@alumni.unitn.it

P. Giorgini
E-mail: paolo.giorgini@unitn.it

J. Mylopoulos
E-mail: john.mylopoulos@unitn.it

by a mere technical failure, but rather was due to several social and organizational vulnerabilities in the overall system design [72], which was exploited by some actors [14, 72]. For instance, some High-Frequency Traders (HFTs) intentionally provided falsified information to manipulate the trading environment and make extra profit out of that. In addition, the lack of coordination among the markets enabled some traders to continue their trading activities during the crash by forwarding their orders to markets that did not halt. This lack of coordination resulted also from IQ related vulnerabilities. However, such failures could be avoided if the IQ requirements of the system-to-be were captured properly during system design [30, 29]. This motivates and underscores the need for analyzing the social and organizational environment where the BP is executed [42].

On the other hand, the techniques mentioned above define "what" mechanisms and solutions are needed to solve IQ problems, but they do not specify "why" such mechanisms and solutions are needed. We advocate that understanding "why" IQ mechanisms and solutions are needed can provide a better understanding stakeholder needs that go beyond IQ requirements. In [31], we proposed a goal-oriented approach for capturing IQ requirements of the overall system where a BP is enacted, which allows identifying "why" a certain level of IQ is needed. Moreover, we proposed a mechanism for mapping these requirements into workflow nets with actors (WFA-net). WFA-net is a workflow language for modeling and analyzing control-flow, information-flow, and IQ requirements of the BP.

In our previous work [31], we proposed an approach to analyze IQ requirement in terms of three IQ dimensions (accessibility, accuracy, and consistency). In this paper, we extend our framework proposing seven IQ dimensions (accessibility, accuracy, completeness, believability, trustworthiness, timeliness, and consistency), and we extend the modeling language that is used to model the overall system. Moreover, we extend the semantics of WFA-nets to model IQ in terms of these seven dimensions, and we refine the mapping constraints and the reasoning techniques to cope with such extensions. In addition, the paper presents a prototype implementation, as well as two realistic examples from the stock market domain.

The paper is organized as follows; Section (§2) presents the research baseline, Section (§3) describes the U.S. stock market system that is used as an example to illustrate our approach. We present and discuss our approach for modeling and analyzing IQ requirements of BPs in section (§4), and in Section (§5) we implement and evaluate the approach. In Section (§6), we discuss limitations of our proposal and threats to validity. We present related work in Section (§7). Finally, we conclude and discuss future work in Section (§8).

2 Research baseline

Information Quality (IQ). IQ is a hierarchical multi-dimensional concept characterized by dimensions such as accessibility, accuracy, completeness [57, 82, 54, 9]. Several models have been proposed for analyzing IQ based on these dimensions (e.g., [45, 54, 9]). However, there is no consensus on what these dimensions should be [45]. Moreover, most of these models ignore social and organizational aspects that may underlie some of these dimensions, which leaves the system open to different kinds of social and organizational vulnerabilities.

In [28], we tackled this problem by proposing a multi-dimensional model (shown in Figure 1) for analyzing IQ that considers social and organizational aspects while analyzing IQ dimensions. The model analyzes IQ based on seven IQ dimensions: accessibility, accuracy, believability, trustworthiness, completeness, timeliness and consistency. These dimensions have been chosen based on the following criteria. Although there is no general agreement on which are the most important IQ dimensions, it is easy to note that four IQ dimensions have been considered in most of the IQ models: accuracy, completeness, timeliness and consistency (e.g., [3, 82, 5, 13]). In addition, we consider both information believability and trustworthiness, since they can be used for analyzing information accuracy [84, 19]. Finally, before thinking about any of the previously mentioned dimensions (e.g., accuracy, completeness, etc.), we need to possess the information and have the required permissions over it to perform a task at hand. Therefore, information accessibility is also considered in our model. In what follows, we define each of these dimensions:

Accessibility measures the extent to which information is available for use [54], i.e., accessibility is defined as information availability along with the required permissions over it to perform a task at hand.

Believability measures the extent to which information is accepted or regarded as true [54, 9].

Trustworthiness measures the extent to which information is credible [45]. *Trustworthiness* can be analyzed based on the *trustworthiness of the provenance*, which can be further analyzed based on both *trustworthiness of the source* and *trustworthiness of the provision* [19].

Accuracy measures the extent to which information is true or error-free with respect to some known or

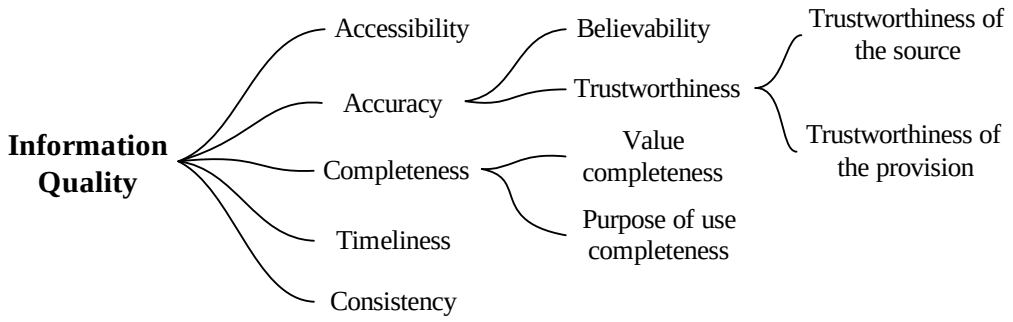

Fig. 1 Multi-dimensional model for analyzing IQ for socio-technical systems

measured value [9]. We analyze accuracy based on the two sub-dimensions *believability* and *trustworthiness*.

Completeness measures the extent to which information is complete for performing a task at hand [9]. We analyze completeness depending on two sub-dimensions, *value completeness*: the extent to which information is preserved against corruption or loss that might endanger its integrity; and *purpose of use completeness*: the extent to which information is complete for performing a task at hand.

Timeliness measures the extent to which information is valid in term of time (e.g., sufficiently up-to-date) for performing a task at hand [54].

Consistency measures the extent to which all multiple records of the same information are the same across time and space [9].

Petri nets, WF-nets, and WFD-nets. Although there exist several BP modeling languages in the literature, we adopt Petri-net based languages (e.g., Petri nets, WF-nets, and WFD-nets) as a baseline of our BP language, since they have a simple, clear and well-defined semantics for process modeling and analysis, and they can be used to model different kinds of BPs.

In particular, a Petri net is a directed graph consisting of two kinds of nodes, places and transitions, where arcs are either from a place to a transition or from a transition to a place [50]. Formally, A Petri net N = $\langle P, T, F\rangle$, where P is a finite set of places, T is a finite set of transitions, and $F \subseteq (P \text{ x } T) \cup (T \text{ x } P)$ is a set of arcs (flow relation). At any time, a place contains zero or more tokens, while a transition $t \in T$ is said to be *enabled*, **iff**, each input place p of t contains at least one token. An *enabled* transition t may *fire*, **iff**, a transition t consumes one token from each input place p of t, and it produces one token in each output place p of t.

Furthermore, a marking of a Petri net is a multi-set of its places $M : P \longrightarrow \mathbb{N}$. Transitions are the active components in a Petri net, i.e., they change the state of the net. For example, given a Petri net N and a marking M_1, we say that $M_1 \xrightarrow{t} M_2$: if transition t is *enabled* at marking M_1, and firing t at M_1 results in M_2. While $M_1 \xrightarrow{\sigma} M_n$: $\sigma = t_1, t_2, \ldots, t_{n-1}$ is a firing sequence leading from M_1 to M_n. Finally, we say that a marking M_n is *reachable* from M_1, **iff**, there is a firing sequence $\sigma = t_1, t_2, \ldots, t_{n-1}$ such that $M_1 \xrightarrow{\sigma} M_n$.

On the other hand, a Workflow net (WF-net) [81] is a subclass of Petri nets intending to model the workflow of process activities, i.e., the WF-net transitions are assigned to tasks or activities, and places are assigned pre/post conditions [81]. In addition, a WF-net is a Petri net with well-defined starting point (**start**) and a well-defined ending point (**end**), and every node (place or transition) is on a path from **start** to **end**. Both of Peti-nets and WF-net have been used to model different kinds of processes from various domains. However, their modeling notation is not expressive enough to capture anything but the control flow of the process.

Finally, Workflow net with data (WFD-net) [68] is a workflow net with data elements, in which tasks can read (rd), write (wt), or delete (del) data elements. Moreover, a task can also have data dependent guards (grd) that block its execution when it is evaluated to false. The authors of WFD-net have proposed an example concerning a shipper's process for delivering goods to illustrate their language, and they showed how their language is able to model the control flow and the information flow of the process. WFD-net allows for identifying "where" information is needed, but not "why" a certain level of IQ is needed.

3 US Stock Market System

A stock market (also called equity market or share market) consists of investors and traders who trade securities[1] at a trading venue (exchange). Kirilenko et al. [41] identify the main stakeholders of a stock market system, including *stock investors*. These are individu-

[1] The term security refers to any tradable financial asset

als or companies, who have as main goal making profit from trading securities in *stock markets*.

Stock traders are individuals or companies involved in trading securities in *stock markets* either for their own sake or on behalf of their *investors* with a main goal of making profit by trading securities. Based on their trading speed and number of orders to be held (traded), *traders* can be broadly classified under three main categories:

High Frequency Traders (HFTs) are traders who trade large volumes of securities with very high frequency (speed) [2].

Market makers are traders who trade large volumes of a particular security on both sides of the market (buy/sell). Their role is usually to facilitate trading on certain security in the market.

Small traders trade small volumes of securities with a low frequency.

Stock markets are places where *traders* gather and trade securities (e.g., New York Stock Exchange (NYSE), Chicago Mercantile Exchange (CME), NASDAQ, etc.). In particular, markets make a profit by facilitating securities trading among traders, i.e., they receive, match, and perform trades among *traders*. Moreover, *markets* should guarantee a fair and stable trading environment for their traders. Usually, they analyze the trading activities they are managing, and employ their Circuit Breakers (CBs), when needed, to slow down or halt trading to prevent a potential market crash. Furthermore, in the stock market system the same security can be traded in several markets, but it will always have only one primary listing market. Therefore, markets need to coordinate their trading activities with the primary market to prevent a market crash.

Consolidate Tape Association (CTA) collects and processes information from *stock markets* concerning their trades and quotes[2], and then disseminates information concerning trades (CTS-info) and quotes (CQS-info) to *traders*/*investors*. Such information enables *traders*/*investors* to analyze the trading environment, and in turn, make the right trading decisions.

On the other hand, Mishkin [48] identifies several other players that provide various services related to the stock market, including: *Consulting firms* that are firms specialized in providing professional advice concerning securities to both *traders* and *investors*, where such advice assists *traders* and *investors* while taking their trading decisions. Finally, *credit assessment rating firms* provide assessments of the credit worthiness of companies' securities, which helps *traders* in deciding how risky it is to trade a certain security.

[2] A quote is an order that has not been performed [36]

The stock market domain is a good example, where IQ is very important for most BPs enacted in this domain. For example, an *investor* depends on trading suggestions to assist its trading decisions, where the quality of such suggestions influences the qualities of the trading decisions an investor may take, and in turn, the success or failure of its trading process.

On the other hand, to make the right trading decisions, *traders* depend on trades and quotes information that is produced by the CTA concerning the quotes that markets receive and trades that markets perform respectively. However, any delay of such information may result in dependence on invalid (outdated) information, which might lead to taking a wrong trading decision, and in turn, the failure of the trading process.

4 Modeling and Reasoning about IQ Requirements in Business Processes

In this section, we present our approach for modeling and analyzing IQ requirements in BP. First, we provide the methodological process that underlies our approach (Section 4.1). Second, we describe the modeling phase that models the IQ requirements of the system in their social and organizational context, where the BP is executed (Section 4.2). Third, we discuss the mapping phase that models the BP of concern by mapping leaf goals of the IQ requirements model into the activities of WFA-net (Section 4.3). Finally, we describe the automated reasoning support that can be used to verify the control-flow, information-flow, and IQ requirements of a BP (Section 4.4).

4.1 The Methodology

The process (shown in Figure 2) is composed of three main phases, namely: modeling, mapping and analysis. In what follows, we briefly describe each of these phases:

(1) **Modeling phase**: aims to model the IQ requirements of the system-to-be in their social and organizational context, where the BP is executed. This phase is composed of five main steps: (1.1) *Actor modeling* aims to model the actors of the system in terms of agents and roles; (1.2) *Goal modeling* aims to model the actors' objectives in terms of goals, and refinements of these goals, if needed, through and/or-decomposition until reaching leaf goals; (1.3) *Information modeling* aims to model the different relations between goals and information (e.g., produce, read, modify and send); the relations between information and its sub-parts; and the relations between legitimate owners of information and information they own. (1.4) *Social dependency modeling*

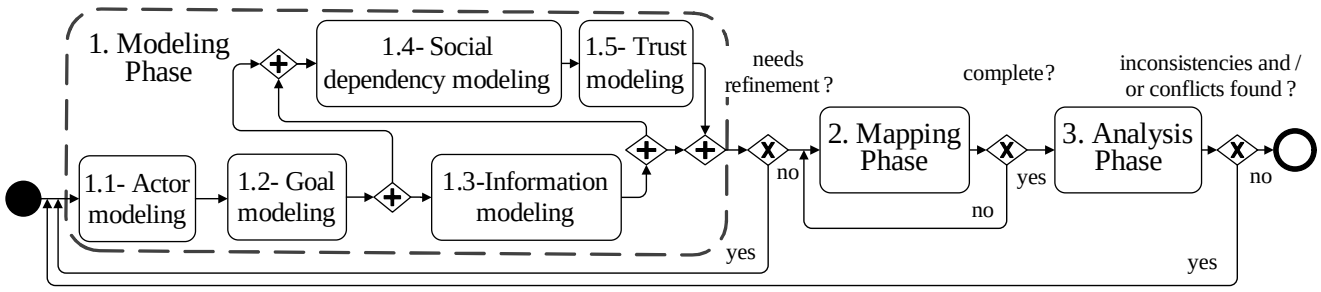

Fig. 2 The process for modeling and reasoning about IQ requirements in BP

aims to model actor dependencies for information provision, and the delegation of both authorities and entitlements among actors, i.e., based on actor capabilities some goals might be delegated to other actors, who have the capabilities to achieve them; and based on actor needs, information/permissions are provided/delegated among them respectively. (1.5) *Trust modeling* aims to model trust/distrust relations among actors concerning information producing/provision, and goal/permission delegations. When the modeling phase is completed, and the model does not require any further refinements, we proceed to the mapping phase.

(2) **Mapping phase**: aims to map the IQ requirements model that has been produced in the previous phase into activities of the WFA-net to represent the intended process taking into consideration the actors who are responsible for achieving such goals, and information that such goals produce, read, modify and/or send. When the mapping phase is completed, we proceed to the analysis phase.

(3) **Analysis phase**: aims to verify the correctness and consistency of the BP model. In particular, we define a set of properties of the design that can be used to verify the correctness and consistency of the control-flow, information-flow, and IQ requirements of the WFA-net, i.e., the WFA-net is correct and consistent, if all of these properties hold.

4.2 Modeling Phase

In order to model the IQ requirements of the system in their social and organizational context, where the BP is executed, we rely on our modeling language proposed in [30, 28], which provides concepts for modeling the system in terms of its actors, goals, IQ requirements, etc. Figure 3 shows a portion of a goal model concerning the stock market system represented with our modeling language to clarify its main concepts. In what follows, first we present the main concepts and constructs for modeling actors, goals, information, along with their different relations and social dependencies. Second, we present the constructs for modeling IQ requirements.

The language introduces constructs for modeling *actors* of the system in terms of *agents* and *roles*. A *role* consists of a set of behaviors and functionalities within some specialized context, and *roles* can be *specialized* from one another [86]. An *agent* is an autonomous entity that can play one or more roles within the system that inherits the properties and behaviors of the *roles* it *plays* [85].

For instance, a `stock trader` has the capability of employing different strategies for making profit out of its trading activities. `HTF` and `Market maker` roles are specialized (through `is_a`) from the `stock trader` role, thereby inheriting the behaviors of the `stock trader` role, and they have additional behaviors that can be used to differentiate them from one another, and from the `stock trader` role. `Best trading` is an agent that `plays` a `stock trader` role.

A *Goal* can be defined as a state of affairs that an actor intends to achieve [10], and it can be refined through *and/or-decompositions* into finer sub-goals. Refining a root-goal into sub-goals through *and-decomposition* implies that all sub-goals need to be achieved in order to achieve the parent goal. While in *or-decomposition*, achieving any of the sub-goals implies achievement of the parent goal. For example, `Best trading` has a main goal of `G1. Make profit by trading securities` that is and-decomposed into `G1.1 Produce the right orders` and `G1.2 Analyze target security`, where achieving `G1` requires achieving both of `G1.1` and `G1.2`. While `G1.2 Analyze target security` is or-decomposed into `G1.2.1 Analyze by itself` and `G1.2.2 Analyze by consultant`, where achieving any of `G1.2.1` or `G1.2.2` is enough for achieving `G1.2`.

Information represents any informational entity, and it has a *volatility* attribute that represents the change rate of its value [84]. Information can be composed of more than one part, and we use the *part of* relation (represented as `partOf`) to capture the relationship between a composite information and its sub parts (e.g., `CME CB info` is *part of* `NYSE CB info`).

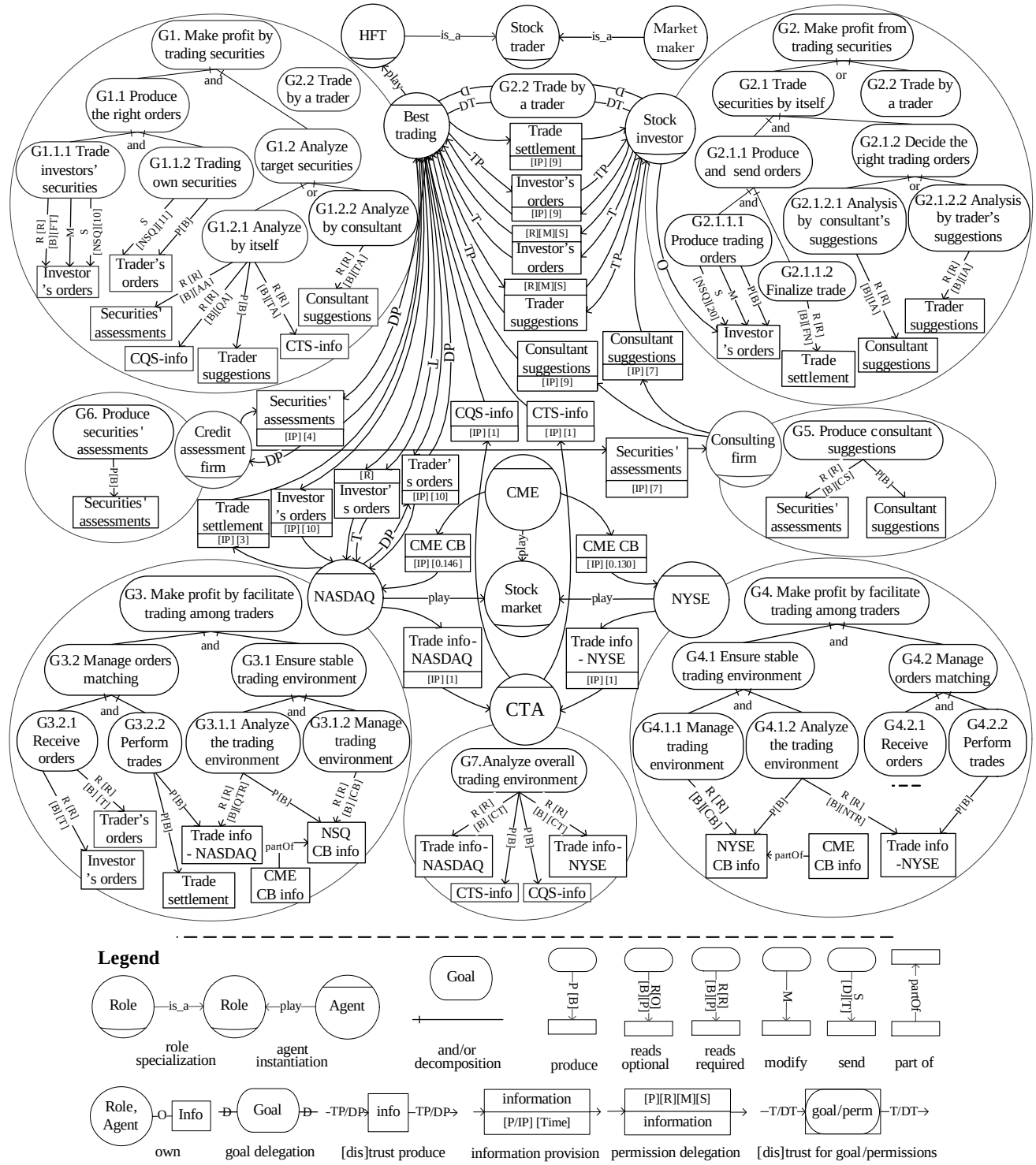

Fig. 3 A partial goal model concerning the stock market structure

Goals may *produce*, *read*, *modify* and/or *send information*. For instance, the goal `G1.2.1` Produces `Trader suggestions`, which indicates that `Trader suggestions` can be created by achieving this goal. Produce relation has one attribute that indicates whether it applies a believability check (represented as `B`) or not (represented as `NB`) while producing information.

The goal `G1.1.1 Reads investor's orders` information, where the first attribute of the read relation is read type that can be strictly classified under `Optional` read, which indicates that information is not required

for goal achievement, i.e., the goal can be achieved even without consuming such information; and Required read, which indicates that information is required for goal achievement, i.e., the goal cannot be achieved without consuming such information. The second attribute in read relation indicates if the read relation applies a believability check or not (B/NB) while reading such information. The third attribute is the purpose-of-use attribute (e.g., **F**orward **T**rades [FT]) that captures the intended purpose of information usage.

Moreover, the goal G1.1.1 Sends investor's orders, where Send relation indicates that the goal achievement depends on transferring information to a specific destination within a predefined time period. Send relation has two attributes, the first is the intended destination (NASDAQ (NSQ)), and the second attribute is the intended sending period (10 seconds). Finally, G1.1.1 Modifies investor's orders, which indicates that goal achievement depends on modifying such information.

Delegation models the transfer of entitlements and responsibilities among actors. For example, the investor delegates the goal G2.2 Trade by a trader to Best trading. On the other hand, *provision* is used to model information communication among actors, and has two attributes: a time attribute that describes the transmission time, and a provision type attribute that can be either Integrity Preserving (IP) provision or normal (P) Provision [25]. For example, the investor provides Best trading with investor's orders, through IP provision within 9 seconds. Moreover, an *actor* can be a legitimate *owner* of information, which gives it full control over information usage. For instance, the investor is the Owner of investor's orders, and it delegates **R**ead, **M**odify, and **S**end *permissions* over investor's orders to Best trading.

Finally, the language adopts the notion of *trust* and *distrust* to capture the actors' expectations of one another concerning their delegated entitlements and authorities [27]. For example, the investor distrusts Best trading for achieving the goal G2.2. While the investor trusts Best trading for **R**ead, **M**odify, and **S**end *permissions* over investor's orders. Moreover, the language introduces trust/distrust for produced information, which indicates that the trustor trusts/distrusts the trustee for producing trustworthy information. Such relation is represented as an edge labeled with TP/DT between the trustor and the trustee concerning the produced information.

After presenting the main constructs for modeling actors, goals, information, along with their different relations and social dependencies, we present the main concepts and constructs for modeling IQ requirements[3] in terms of its seven dimensions:

Information accessibility: is influenced by:

1. Information availability, which is analyzed depending on information provision between information consumers and information sources (producers).
2. Permissions over information, which enables or prevents an actor from using information as intended. For example, Best trading needs to read, modify and send investor's orders, and it will not be able to perform any of these activities unless investor's orders have been provided to it (availability), and Read, Modify and Send permissions over the orders have been delegated to it.

Information completeness: is influenced by:

1. *Value completeness*, whether information has been preserved against loss or corruption during its transfer, i.e., if information is provided through Integrity Preserving **IP** provision [25], its value completeness is guaranteed, otherwise it is not. For example, if CME CB info was provided to NYSE through normal **P** provision, such information will be considered value incomplete by NYSE.
2. *Purpose-of-use completeness*, whether information is complete for performing a task at hand, i.e., information has all required sub-parts for performing the task at hand. For example, if CME CB info was not provided to NYSE, NYSE CB info will be considered incomplete for the purpose-of-use, since CME CB info is part of NYSE CB info.

Information timeliness (validity): the only two relations between goals and information that can be influenced by time-related aspects are read and send. Therefore, we analyze timeliness in both of them as follows:

1. *Read timeliness*, can be analyzed by comparing information *read time* that enables for determining the *currency (age)* of information with information volatility, i.e., information is valid for read if its currency is smaller than its volatility, otherwise it is invalid. For example, Best trading should verify that the currency of investor's orders is bigger than their volatility to guarantee their validity for read.
2. *Send timeliness*, can be analyzed by comparing information *send* time and the *read time* of such information at its destination, i.e., information is valid

[3] Details about capturing IQ requirements at a high-level of abstraction as soft goals and gradually refined and then approximated into IQ constraints (IQC) can be found in [28]

if its *read time* at its destination is smaller than its *send time*, otherwise it is invalid. For example, the `investor` should verify that the provision time of his order to the stock market (NASDAQ) through `Best trading` is less than their send time, to guarantee their validity at the market.

Information consistency: arises only when there are multiple records of the same information that are being read by actors for *interdependent purposes*. We rely on the *purpose-of-use* attribute in read relation to identify *interdependent readers* that are actors who read the same information for the same *purpose-of-use*. In this context, consistency among *interdependent readers* can be analyzed based on the *read times* of their information, i.e., information is consistent among them, if all of them have the same *read time*, otherwise it is inconsistent. For example, `NYSE` and `NASDAQ` read `CME CB info` for the same purpose of use (CB). Therefore, `NYSE` and `NASDAQ` are *interdependent readers* for `CME CB info`, i.e., both of them should have the same read time to guarantee that `CME CB info` is consistent between them.

Information believability: believability concerns arise when information is being produced or read. Therefore, we analyze believability in produce/read relations by checking whether such relations apply a believability check, i.e., the produced/read information is believable from the perspective of its producer/reader, if the produce/read relation applies a believability check, otherwise it is not. For example, `Trader suggestions` and `CTS-info` are believable, since the goal `G1.2.1` applies a believability check while producing and reading each of them respectively.

Information trustworthiness: is influenced by

1. *Trustworthiness of the source*, which can be analyzed depending on trust/distrust of produce between the trustor and the trustee concerning the produced information. For example, `Best trading` trusts the `stock investor` for producing `investor's orders`.
2. *Trustworthiness of the provision*, which can be analyzed based on the way information arrives at its final destination [11, 69], taking into consideration the operations that have been applied to it, whether the actor who performs such operations is authorized, and whether such authorization is trusted. For example, `NASDAQ` analysis the trustworthiness of provision concerning `investor's orders` by checking whether such information has been modified by `Best trading`, whether `Best trading` has the modify permissions over it, and whether such permission is trusted by the `investor` (information owner).

Information accuracy: following [19, 84], we analyze accuracy depending on *believability* and *trustworthiness*, and we differentiate between two cases:

1. *Accuracy of produced information*, can be analyzed based on the *believability* of the produce relation, and the *trustworthiness of the production process*. For example, if the `investor` depends on `Best trading` for producing its orders, to analyze the accuracy of the produced orders, we need to check whether the produce relation applies a believability check, and whether there is a trust relationship between the `investor` and `Best trading` concerning the produce permission.
2. *Accuracy of read information*, can be analyzed based on the *believability* of read relation, and the *trustworthiness of the provenance*. For example, to analyze the accuracy of an order that a stock market reads, we need to check whether the read relation applies a believability check, and we need to analyze the trustworthiness of the order provenance, which can be done as described earlier.

4.3 Mapping phase

In this section, we extend the semantics of WFA-net to model and analyze IQ requirements in terms of seven IQ dimensions, and then we discuss the mechanisms that we use for mapping IQ requirements model into WFA-net.

4.3.1 Workflow net with Actors (WFA-net)

A workflow net with Actors (WFA-net) adopts workflow net (WF-net), and extends it with the notion of social actor, and IQ needs. In WFA-net, each activity (transition) is assigned a social actor, and it may produce, read, modifies, and sends information. In what follows, we define the semantics of WFA-nets. Let us consider a finite set of social actors $A = \{a_1, a_2, \ldots, a_n\}$, a finite set of information $I = \{i_1, i_2, \ldots, i_m\}$, a finite set of time intervals $T = \{t_1, t_2, \ldots, t_m\}$, and we define $I_v \subseteq \{I \text{ x } T\}$ a finite set of information along with their volatility values.

Moreover, to capture information produce relation, we define $P \subseteq \{Bv \text{ x } I_v\}$, where $Bv = \{B, NB\}$ is the believability check mechanism that is/is not applied by the produce relation, and $i \in I_v$ is information to be produced. To capture information read relation, we define $R \subseteq \{Rt \text{ x } Bv \text{ x } PoU \text{ x } I_v\}$, where $Rt = \{o, r\}$

is the read type that can be either optional or required, $Bv = \{B, NB\}$ is the believability check mechanism that is/is not applied by the read relation, PoU is the purpose for which information is read, and I_v is the information to be read. To capture information send relation, we define $S \subseteq \{A \text{ x } I_v \text{ x } T\}$, where A is a set of social actors that represent the intended destination of information, I_v information to be sent, and T the required time period for information to be sent.

Furthermore, we define a set of responsibility predicates $\Pi_A = \{\pi_{a_1}, \pi_{a_2}, \ldots, \pi_{a_k}\}$ to capture the relations between activities and actors, who are responsible for their execution (e.g., res(actor: a)); a set of produce predicates $\Pi_P = \{\pi_{p_1}, \pi_{p_2}, \ldots, \pi_{p_j}\}$ to capture the relation between activities and information they produce (e.g., produces(blvType: bt, information: i_v)); a set of read predicates $\Pi_R = \{\pi_{r_1}, \pi_{r_2}, \ldots, \pi_{r_k}\}$ to capture the relation between activities and information they read (e.g., read(type: r/o, blvType: bt, PoU: pou, information: i_v)); a set of modify predicates $\Pi_M = \{\pi_{m_1}, \pi_{m_2}, \ldots, \pi_{m_l}\}$ to capture relations between activities and information they modify (e.g., modify(information: i_v)); and a set of send predicates $\Pi_S = \{\pi_{s_1}, \pi_{s_2}, \ldots, \pi_{s_l}\}$ to capture the relation between activities and information they send (e.g., send(actor: a, information: i_v, time: t)).

In addition, we define a responsibility assigning function, $f_{\pi_a} = \Pi_A \longrightarrow \{A\}$ that assigns responsibility predicates with actors that are responsible for activity execution; a produce assigning function, $f_{\pi_p} = \Pi_P \longrightarrow 2^P$ that assigns produce predicates with information that activities produce; a read assigning function $f_{\pi_r} = \Pi_R \longrightarrow 2^R$ that assigns read predicates with information that activities read; a modify assigning function $f_{\pi_m} = \Pi_M \longrightarrow 2^{I_v}$ that assigns modify predicates with information that activities modify; and a send assigning function $f_{\pi_s} = \Pi_S \longrightarrow 2^S$ that assigns send predicates with information that activities send.

To this end, we define a WFA-net as a WF-net, where each activity t is described by, an actor that is responsible (res) for the activity, a set of information that activity produces (pd), a set of information that activity reads (rd), a set of information that activity modifies (md), and a set of information that activity sends (sd).

Definition 1 (WFA-net) A workflow net with actors (WFA-net) N = ⟨ P, T, F, res, pd, rd, md, sd ⟩ consists of a WF-net N= ⟨ P, T, F ⟩, a responsibility assigning function res: T $\longrightarrow A$, a produce assigning function pd: T $\longrightarrow 2^P$, a read assigning function rd: T $\longrightarrow 2^R$, a modify assigning function md: T $\longrightarrow 2^{I_v}$, and a send assigning function sd: T $\longrightarrow 2^S$.

Example 1 a WFA-net of a stock investor that is trading securities is shown in Figure 4. The actor set A = {`investor`, `Best trading`, `NYSE`, `NASDAQ`, `CTA`, `credit assessment firm`, `consulting firm`}, and its information set I_v= {`trade info-NYSE`, `trade info-NASDAQ`, `CTS-info`, `CQS-info`, `trade settlement`, `securities' assessment`, `trader suggestions`, `consultant suggestion`, `investor's orders`}. Considering the activity `T10: Produce trading orders`, the responsibility assigning function `res`(Produce trading orders) = {`investor`}, the produce assigning function `pd`(Produce trading orders) = {`investor's order`}, the read assigning function `rd`(Produce trading orders) = {∅}, the modify assigning function `md`(Produce trading orders) = {`investor's order`}, and the send assigning function `sd`(Produce trading orders) = {(`NASDAQ`, `investor's order`, `20`)}.

On the other hand, to capture the work flow in WFA-net, we should be able to evaluate the activities related predicates either to true (⊤) or to false (⊥). Therefore, we define the following functions, σ_{π_a}: $\Pi_A \longrightarrow \{\top, \bot\}$ assigns to each responsibility predicate either ⊤, when the responsible actor can and will achieve the activity[4], or it assigns ⊥ otherwise. Similarly, we define σ_{π_p}: $\Pi_P \longrightarrow \{\top, \bot\}$, σ_{π_r}: $\Pi_R \longrightarrow \{\top, \bot\}$, σ_{π_m}: $\Pi_M \longrightarrow \{\top, \bot\}$, and σ_{π_s}: $\Pi_S \longrightarrow \{\top, \bot\}$ that assigns to each produce/read/modify/send predicate either ⊤, when information i can be produced/read/modified/sent by the activity, or it assigns ⊥ otherwise. Finally, we define activity state σ: $T \rightarrow$ (⊤, ⊥) that sums the values of all the previously mentioned functions over their related predicates for a specific activity, and Σ denotes the set of all activity states. In this context, a WFA-net configuration consist of a **marking**[5] m along with an **activity state** σ.

Definition 2 (Configuration of WFA-net) Let N = ⟨ P, T, F, res, pd, rd, md, sd ⟩ be a WFA-net, let m be a **marking** of N, and let $\sigma \in \Sigma$ be as defined above. Then, c = $\langle m, \sigma \rangle$ is a configuration of N. With Ξ we denote the set of all configurations of N, and the start configuration of N is defined by $c_s = \{\langle [start] , \sigma \rangle, \mid \sigma \in \Sigma, I_v = \emptyset\}\rangle$. While $C_e = \{\langle [end], \sigma \rangle \mid \sigma \in \Sigma, I_v\}$ defines the set of final configurations.

In the initial configuration, only one place is marked [start], and the I_v set is initialized to the empty set, since no information has been produced yet. While a configuration is a final configuration, if it contains a marking [end].

[4] The responsible actor has the capability, and we trust it for achieving such activity

[5] A marking of a Petri net is a distribution of tokens over its places

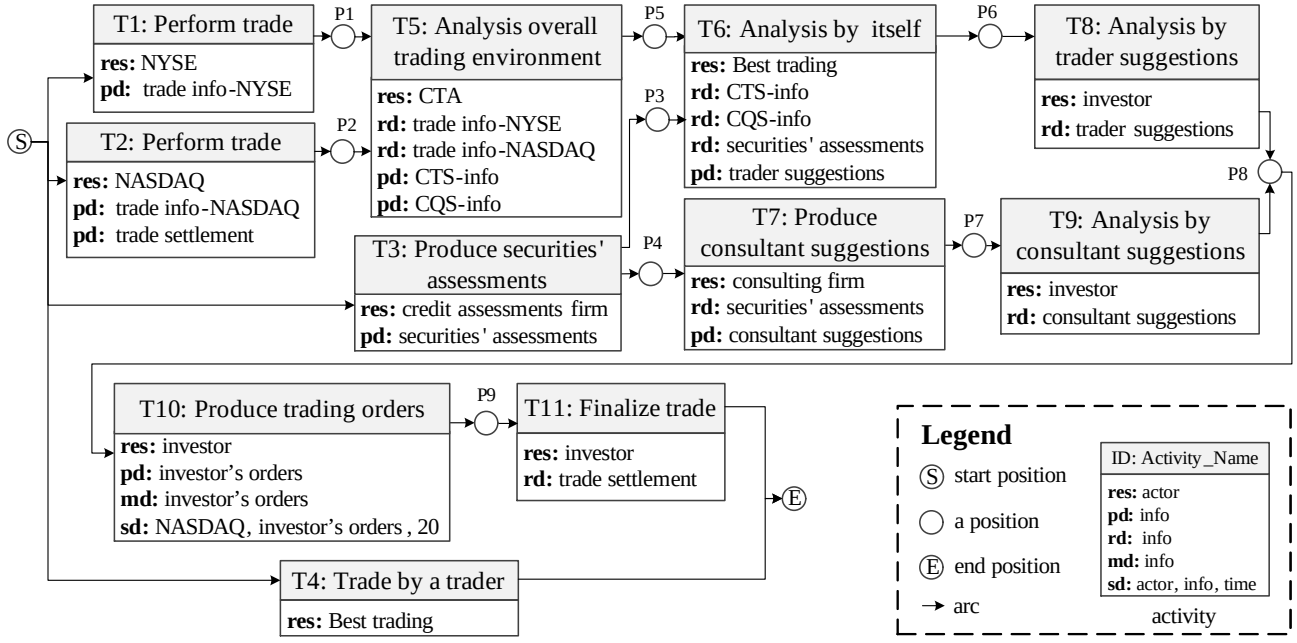

Fig. 4 A WFA-net concerning a stock investor process for trading securities

An activity t of a WFA-net N can be enabled at a configuration c $= \langle m, \sigma \rangle$, **iff**: (1) the activity t is enabled at marking m (activity flow), and (2) the activity state (σ) is evaluated true ($\top$), i.e., information-flow and IQ requirements (if any) are met. When an activity is enabled, it may fire, where firing of an activity changes the marking as well as the activity state and information set I_v, i.e., the firing of an activity enables a set of successor configurations $\langle \mathrm{m}^{'}, \sigma^{'} \rangle$, and it may change the information set as well.

Definition 3 (Firing an activity of WFA-net) Let N $= \langle$ P, T, F, res, pd, rd, md, sd $\rangle$ be a WFA-net. An activity $t \in$ T of N is enabled at a configuration c $= \langle m, \sigma \rangle$ of N if $m \xrightarrow{t}$, σ is assigned true ($\top$). The firing of t enables a set of configurations $c \xrightarrow{t} C \subseteq \Xi$. C $= \{\langle \mathrm{m}^{'}, \sigma^{'} \rangle \mid m \xrightarrow{t} m^{'} \wedge (\forall\, i \in \mathrm{pd}(t) = \top\colon I_v^{'} = I_v \cap i)\}$.

Example 2 consider activity `T10: Produce trading orders` in Figure 4, and suppose there is a token in place `P8`. The activity is enabled if (1) `investor` (responsible actor) is capable and will achieve such activity, (2) producing `investor's orders` will not be prevented, i.e., *investor* is allowed to produce `investors orders`, and such information is accurately produced from its perspective (the produce relation applies a believability check while producing such information), (3) modifying `investor's orders` will not be prevented, i.e., `investor` is allowed to modify such information, and (4) sending `investor's orders` to NASDAQ within 20 seconds will not be prevented, i.e., `investor's orders` will be accurate, complete, and valid at its destination. Firing this activity means that the token in `P8` will be removed, and a token at `P9` will be produced. Moreover, information `investor's orders` will be added to the information set I_v, since it has been produced.

A configuration $c^{'}$ may need more than one transition to be *reached* from another configuration c. Therefore, we need to extend the firing of a single transition to firing a sequence of transitions in order to define the set of reachable configurations of N from a configuration c. In particular, *reachability* is used to find whether there is a path that a configuration $c^{'}$ can be reached from a configuration c, which can be done by proceeding from configuration c through several configurations to find the requested configuration (e.g., $c^{'}$). In such case, we either reach $c^{'}$ from c, or we say that $c^{'}$ cannot be reached from c.

Definition 4 (Reachability of a configuration of WFA-net) Let N $= \langle$ P, T, F, res, pd, rd, md, sd $\rangle$ be a WFA-net, c and $c^{'}$ are two configurations of N. We say that a configuration $c^{'}$ is reachable from a configuration c, denoted by $c \xrightarrow{t_n} c^{'}$, if there is a firing sequence $t_n = \{t_1, \ldots, t_n\} \in$ T being enabled at $c_1, \ldots, c^{'}$, and the firing of t_n leads to $c^{'}$.

Example 3 Figure 5 shows all possible configurations concerning a stock investor process for trading securities that is represented in Figure 4. Each configuration is represented as an oval that is described with an identifier and a marking. For instance, configuration $c1$ consists of a marking [p1], and from the start configuration

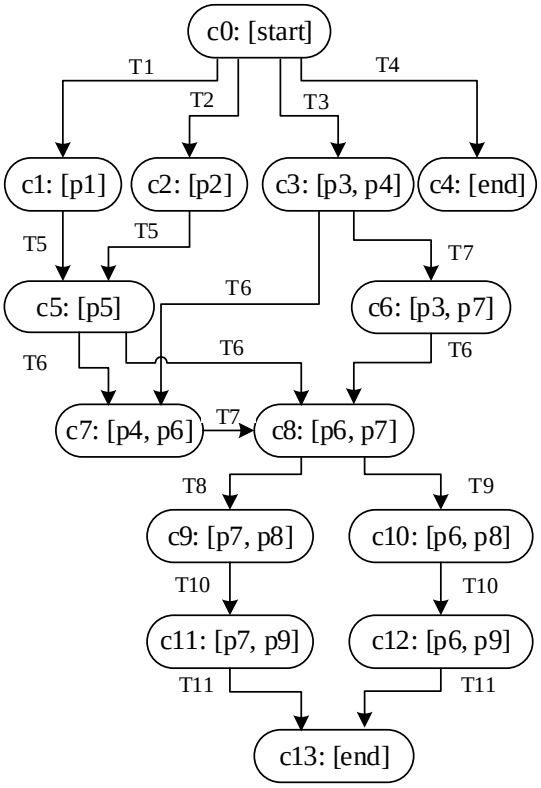

Fig. 5 All possible configurations concerning a stock investor process for trading securities

$c0$, configurations $c1-4$ can be reached, i.e., $c0 \longrightarrow \{c1, c2, c3, c4\}$,

The *soundness* property in WFA-net is used to verify whether the final configuration is reachable from any configuration of N.

Definition 5 (Soundness of WFA-net) Let N = ⟨ P, T, F, res, pd, rd, md, sd ⟩ be a WFA-net, and c_0 be the start configuration of N, C ⊆ Ξ is a set of configurations of N, and $c_e \subseteq \Xi$ is a set of final configurations of N. We say N is sound iff, for every configuration c reachable from c_0, there exists a firing sequence leading from c to c_e, i.e., $\forall c \in C : (c_0 \xrightarrow{*} c) \Rightarrow (c \xrightarrow{*} c_e)$.

Example 4 considering Figure 5, $c0$[**start**] $\longrightarrow \{c1[P1], c2[P2], c3[P3], c4$[**end**]$\}$, $c1 \longrightarrow c5[P5]$, $c2[P2] \longrightarrow c5[P5]$, $c5[P5] \longrightarrow \{c7[P4, P6], c8[P6, P7]\}$, $c3[P3, P4] \longrightarrow \{c6[P3, P7], c7[P4, P6]\}$, $c6[P3, P7] \longrightarrow c8[P6, P7]$, $c7[P4, P6] \longrightarrow c8[P6, P7]$, $c8[P6, P7] \longrightarrow \{c9[P7, P8], c10[P6, P8]\}$, $c9[P7, P8] \longrightarrow c11[P7, P9]$, $c10[P6, P8] \longrightarrow c12[P6, P9]$. Finally, $c11[P7, P9] \longrightarrow c13$[**end**], and $c12[P6, P9] \longrightarrow c13$[**end**]. It is clear that from any configuration that can be reached from the initial configuration; the final configuration can be reached. Therefore, the workflow is sound.

4.3.2 Mapping IQ Requirements into WFA-nets

In this section, we describe how the IQ requirements model can be mapped into WFA-net. First, we define rules for identifying building blocks, and then we define three sets of constraints that should be followed during the mapping process to guarantee the correctness of both the mapping and the resulting WFA-net:

Building blocks: are used to represent constructs (goals) of the requirements model, which can be mapped into activities of WFA-net. We define three rules for identifying building blocks that are used to guarantee the correct mapping:

1. A goal that is not and/or-decomposed of any other goal, and it is not decomposed into sub-goals as well, is considered as a building block, which can be mapped into an activity of WFA-net taking into consideration the actor, who is responsible for its achievement, and information that is produced, read, modified and/or sent (if any).
2. A goal that is and-decomposed into sub-goals is considered as a building block in terms of all its sub-goals, which can be mapped into sequencing activities of the WFA-net, where each of these activities represents a sub-goal. The mapping of and-decomposed sub-goals into sequencing activities is derived from the semantics of and-decomposition relation, which implies that all and-decomposed sub-goals should be achieved to achieve the parent goal, and mapping such goals into sequencing activities implies that all of them should be achieved in the WFA-net.
3. A goal that is or-decomposed into sub-goals, is considered as a building block in terms of its all sub-goals which can be mapped into parallel (alternatives) activities of the WFA-net, where each of these activities represents a sub-goal. The mapping of or-decomposed sub-goals into parallel activities is derived from the semantics of or-decomposition relation, which implies that any of the or-decomposed goals should be achieved to achieve the parent goal, and mapping such goals into parallel activities implies that it is enough to achieve any one of them in the WFA-net.

Example 5 consider the IQ requirements model in Figure 3, the goal `G2.2 Trade by a trader` in the scope of `Best trading` can be considered as a building block, since it is not and/or-decomposed of any other goal, and it is not decomposed into sub-goals as well. On the other hand, the goal `G2.1.1 Produce and send orders` is and-decomposed into two sub-goal `G2.1.1.1 Produce trading orders` and

G2.1.1.2 Finalize trade. Therefore, G2.1.1 can be considered as a building block in terms of these two sub-goals, which can be mapped into two sequencing activities in the WFA-net. While the goal G2.1.2 Decide the right trading orders is or-decomposed into two sub-goals G2.1.2.1 Analysis by consultant's suggestions and G2.1.2.2 Analysis by trader's suggestions. Thus, G2.1.2 can be considered as a building block in terms of these two sub-goals, which can be mapped into two parallel activities in the WFA-net.

Consistency constraints: we define three consistency constraints that are used to guarantee a correct mapping between building blocks and activities of the WFA-net:

1. Mapping is allowed for building blocks only, i.e., no goal is allowed to be mapped unless it can be considered as a building block. This constraint prevents designers from arbitrarily mapping goals/sub-goals into activities of the WFA-net.
2. Mapping is allowed for leaf goals only, i.e., no goal is allowed to be mapped unless it is a leaf goal, i.e., it is not and/or-decomposed into sub-goals.
3. No information is allowed in the WFA-net unless its source (the goal/activity that produces such information) exists in the WFA-net. Including information sources in the WFA-net enables for analyzing information availability and several IQ dimensions (e.g., accuracy, completeness, etc.).

Example 6 consider the previous example, the goal G2.2 Trade by a trader can be mapped into an activity of WFA-net, since it is a building block. While the goal G2.1.1 is allowed to be mapped in terms of its two sub-goals G2.1.1.1 and G2.1.1.2 as sequencing activities of the WFA-net, since both of them are leaf goals. In case they were not leaf goals, each of them can be mapped in terms of its sub-goals. On the other hand, consider the WFA-net that is shown in Figure 4, activity T5: Analysis overall trading environment needs to read both of trade info-NYSE and trade info-NASDAQ that is why activities T1 and T2 have been added to the WFA-net, since such activities produce the information required by T5.

Sequencing constraints: we define two sequencing constraints that are used to guarantee the proper ordering of the activities of WFA-net:

1. Activities of WFA-net should be consistent with their sequencing order in their own building blocks.
2. If an activity depends on the outcome of another activity (e.g., information), it should appear after the activity it depends on if possible.

Example 7 consider the previous example, goals G2.1.1.1 and G2.1.1.2 should be mapped into sequencing activities of WFA-net. While the goals G2.1.2.1 and G2.1.2.2 should be mapped into parallel activities of WFA-net. The designer can either map goals G2.1.1.1 and G2.1.1.2 as sequencing activities followed by goals G2.1.2.1 and G2.1.2.2 as parallel activities, or he/she can map goals G2.1.2.1 and G2.1.2.2 as parallel activities followed by goals G2.1.2.1 and G2.1.2.2 as sequencing activities. But it is not allowed to map such goals in any other order, i.e., it is not allowed to separate between goals G2.1.1.1 and G2.1.1.2 that are mapped as sequencing activities by the goals G2.1.2.1 and G2.1.2.2. On the other hand, consider the WFA-net that is shown in Figure 4, activity T5 needs to read both of trade info-NYSE,which is produced by tasks T1 that is why T1 appears before T5 in the WFA-net.

Refinement constraints: we define two refinement constraints that are derived from the semantics of WFA-net, and they are used to guarantee the correct sequencing of the activities of WFA-net that resulted after applying the sequencing constraints.

1. No two places can appear in sequence without an activity separating them;
2. No two activities can appear in sequence without a place separating them.

Example 8 consider the WFA-net that is shown in Figure 4, if there is a position separating T1 and P1, it should be removed. On the other hand, if there was no position separating T1 and T5, a position should be added to separate between them.

An illustrative example: in this example, we show how the investor process for achieving G2. make profit from trading securities shown in Figure 3 can be mapped into WFA-net shown in Figure 4.

The investor aims for achieving the top-level goal G2, but it cannot be considered as a building block since it is or-decomposed into G2.1 and G2.2. Therefore, instead of G2, we have G2.1 and G2.2 that should be represented as parallel activities. G2.2 can be mapped into activity T4, since it is a leaf goal. While G2.1 is and-decomposed into G2.1.1 and G2.1.2, which should be represented as two sequential activities. Where G2.1.1 is also and-decomposed into G2.1.1.1 and G2.1.1.2, which can be mapped into two sequential transactions T10 and T11 respectively. While G2.1.2 is or-decomposed into G2.1.2.2 and G1.1.2.1, and they can be mapped into two parallel activities T8 and T9 respectively.

Activity T8 needs to read trader suggestions, and activity T9 needs to read consultant suggestions, which are produced by goal G1.2.1 Analyze by itself and goal G5. Produce consultant suggestions respectively. Since no information is allowed to exist in WFA-net without its source, goals G1.2.1 and G5 are mapped as activities T6 and T7 activities that produce trader suggestion and consultant suggestion respectively.

However, T6 requires to read both of CTS-info and securities' assessment, which are produced by goal G7. Analysis overall trading environment and goal G6. Produce securities' assessments respectively. Therefore, the goals G7 and G6 are mapped as activities T5 and T3 respectively. Similarly, T7 requires to read securities' assessment that is produced by goal G7, which has been already mapped as an activity T3. Thus, we only map an arc from T3 to T7. Moreover, activity T5 requires to read both trade info-NYSE and trade info-NASDAQ, which are produced by goal G4.2.2 Perform trades and goal G3.2.2 Perform trades respectively. Thus, goals G4.2.2 and G3.2.2 are mapped as activities T1 and T2 respectively.

At this point, we check whether the refinement constraints are respected, i.e., if there is missing a position between activities, or a position separating a position and an activity, we modify the WFA-net accordingly. Finally, T1, T2, and T4 do not need any preceding activities, thus, the Start position is linked directly to them. On the other hand, T4 and T11 do not have any succeeding activities, therefore, they are linked to the End position.

4.4 Analysis Phase

After completing the mapping phase, we have the WFA-net that represents the BP we desire. However, we cannot rely on the WFA-net model to perform any kind of automated analysis without a formal representation of its semantics. Therefore, we provide disjunctive Datalog [8] formalization of all the concepts that are used to model the WFA-net and the IQ requirements, which enables for transforming all constructs of the graphical model (e.g., actor, goal, etc.) into their corresponding formal predicates. Moreover, we adopt DLV system[6] as an inference engine that allows for deducing new knowledge (facts) from the predicates that have been derived from the graphical model based on already defined reasoning axioms (rules)[7], and for performing the required analysis to verify the WFA-net model. Note that we mainly rely on predicates that are derived from the requirements model for analyzing the IQ requirements in WFA-net. While we mainly rely on predicates derived from WFA-net for analyzing the control-flow and information-flow of the WFA-net.

In addition, we define a set of properties of the design (shown in Table 1), which specify logical constraints that the designers should consider during the system design, and they are used to verify the correctness of the mapping, control-flow, information-flow and IQ requirements of the WFA-net model. In what follows, we discuss each of these properties:

Pro1-6 are used to verify the mapping properties of the WFA-net, where **Pro2-6** are derived from the semantics of the WFA-nets, and they are specialized for verifying whether every activity and every position are on a path between the Start and End positions.

Pro1 states that only leaf goals are allowed to be mapped as activities of WFA-net. Considering Figure 3, if goal G.1, goal G.1.1, goal G.1.2, or any other non-leaf goal has been mapped into an activity of WFA-net, *Pro1* will notify the designer that such goal cannot be mapped into an activity since it is not a leaf goal.

Pro2 states that any activity of a WFA-net that has an outgoing arc, should have at least one incoming arc. Consider T(9) in Figure 4 for example, if T(9) has an outgoing arc to a position (e.g., P(8)), and there is no incoming arc from a position (e.g., P(7)), *Pro2* will notify the designer that T(9) have an outgoing arc, but it does not have an incoming arc.

Pro3 states that any activity of a WFA-net that has an incoming arc, should have at least one outgoing arc. Consider T(8) in Figure 4 for example, if T(8) has an incoming arc from a position (e.g., P(6)), and there is no outgoing arc to a position (e.g., P(8)), *Pro3* will notify the designer that T(8) have an incoming arc, but it does not have an outgoing arc.

Pro4 states that the Start position in a WFA-net should be connected with at least one activity. Consider the position P(S) in Figure 4 for example, if there is no outgoing arc from P(S) to at least one activity (e.g.,T(1), T(2), etc.), *Pro4* will notify the designer that position P(S) is not connected properly in the WFA-net.

Pro5 states that any position (not P(S) or P(E) positions) in a WFA-net should be connected with at least two activities through one incoming and one outgoing arcs. Consider the position P(1) in Figure 4 for

[6] http://www.dlvsystem.com/

[7] The formalization of the concepts and axioms is omitted due to space limitation, yet they can be found at https://mohamadgharib.wordpress.com/bpsts-iq-tool/

Table 1 Properties of the design

Mapping properties	
Pro1	:- activity(G), not_leaf(G).
Pro2	:- incoming_arc(G), not outgoing_arc(G).
Pro3	:- outgoing_arc(G), not incoming_arc(G).
Pro4	:- start(P), not starting_arc(P).
Pro5	:- between(P), not connected(P).
Pro6	:- end(P), not ending_arc(P).
Information flow property	
Pro7	:- wf_reads(G, I), not wf_produced(I).
Information Quality properties	
Pro8	:- is_responsible(A, G), activity(G), produce(Type, G, I, T), not has_perm(produce, A, I).
Pro9	:- is_responsible(A, G), activity(G), produce(Type, G, I, T), not accurate_produce(A, I).
Pro10	:- is_responsible(A, G), activity(G), read(T, P, BT, G, I), not has_perm(read, A, I).
Pro11	:- is_responsible(A, G), activity(G), read(T, P, BT, G, I), not accurate_read(A, I).
Pro12	:- is_responsible(A, G), activity(G), read(T, P, BT, G, I), not valid_read(A, I).
Pro13	:- is_responsible(A, G), activity(G), read(T, P, BT, G, I), not complete_read(A, I).
Pro14	:- is_responsible(A, G), activity(G), read(T, P, BT, G, I), not consistent_read(A, I).
Pro15	:- is_responsible(A, G), activity(G), modify(G, I), not has_perm(modify, A, I).
Pro16	:- is_responsible(A, G), activity(G), send(T, G, B, I), not has_perm(send, A, I).
Pro17	:- is_responsible(A, G), activity(G), send(T, G, B, I), has(B, I), not accurate_send(T, A, B, I).
Pro18	:- is_responsible(A, G), activity(G), send(T, G, B, I), not complete_send(T, A, B, I).
Pro19	:- is_responsible(A, G), activity(G), send(T, G, B, I), not valid_send(T, A, B, I).
Control flow properties	
Pro20	:- wf_prevented(G).
Pro21	:- not reached(end).

example, if P(1) does not have an incoming arc from an activity (e.g.,T(1)) and/or it does not have an outgoing arc to an activity (e.g., T(5)), *Pro5* will notify the designer that position P(1) is not connected properly in the WFA-net.

Pro6 states that the End position in a WFA-net should be connected with at least one activity. Considering P(E) position in Figure 4, if there is no incoming arc from at least one activity (e.g.,T(4), T(11), etc.) to P(E), *Pro6* will notify the designer that position P(E) is not connected properly in the WFA-net.

Pro7 states that any activity of WFA-net should have all information it requires (e.g., read, modify, send), where this property is used to verify information availability (information-flow) for activities of a WFA-net. Consider T(9) in Figure 4 for example, T(9) need to read consultant suggestions, if such information has not been produced and provided to the investor that is the responsible actor for achieving activity T(9), *Pro7* will notify the designer that consultant suggestions is unavailable for investor for achieving T(9).

Pro8-19 are used to verify IQ related properties of the activities of a WFA-net, where these properties have been derived from the satisfaction semantics of IQ requirements. In particular, IQ requirements are transformed into design constraints that when respected the IQ requirements are satisfied. For instance, **Pro8** states that a WFA-net should not include any activity that produces information, and the actor who is responsible for achieving such activity does not have a produce permission concerning such information. Consider activity T(10) in Figure 4 for example, T(10) produces investors' orders. If the investor (responsible actor) does not have a produce permission, *Pro8* will detect such violation and notify the designer that such information cannot be produced, since the responsible actor does not have a produce permission.

Pro9 states that a WFA-net should not include any activity that produces inaccurate information from the perspective of the actor who responsible for achieving such activity, where produced information is accurate, if its believability and the trustworthiness of production have been verified. Considering activity T(10), the investor is the legitimate owner of such information, i.e., the trustworthiness of production is verified. Moreover, the produce relation applies a believability check while producing such information (shown in Figure 3). If the trustworthiness of the production is not verified and/or no believability check has been applied, *Pro9* notify the designer that such information is not accurate.

Pro10 states that a WFA-net should not include any activity that reads information, and the actor who is responsible for achieving such activity does not have a read permission concerning such information. Considering activity T(8), if the investor does not have a read permission concerning trader suggestions, *Pro10* will detect such violation and notify the designer that such information cannot be read, since the responsible actor does not have a read permission.

Pro11 states that a WFA-net should not include any activity that reads information, and such information is inaccurate from the perspective of the actor (reader) who is responsible for the activity achievement. Information is accurate from the perspective of its reader, if its believability and trustworthiness of provenance have not been verified. For example, activity `T(8)` reads `trader suggestions`, as shown in Figure 3 the `investor`, who is responsible for achieving `T(8)` trusts `Best trading` (information producer) for producing such information (trustworthiness of the source), and `trader suggestions` has been provided to `investor` without being inappropriately modified (trustworthiness of the provision), thus, the trustworthiness of provenance is verified. Moreover, the goal `G2.1.2.2 Analysis by trader's suggestions` that is mapped as `T(8)` applies a believability check while reading such information, i.e., the believability of such information is verified. In case, the believability and/or the trustworthiness of provenance of `trader suggestions` have not been verified, *Pro11* will notify the designer that such information is inaccurate for read.

Pro12 states that a WFA-net should not include any activity that reads information, and such information is invalid from the perspective of the actor who is responsible the activity achievement. Information is valid for read if its currency (age) is smaller than its volatility, otherwise it is invalid. Considering activity `T(8)`, if the provision time of `trader suggestions` to the `investor` is bigger than the volatility of such information, *Pro12* will notify the designer that such information is invalid for read.

Pro13 states that a WFA-net should not include any activity that reads information, and such information is incomplete from the perspective of the actor who is responsible for the activity achievement. Information is complete for read, if it is value complete (information has been preserved against lost and corruption during its transfer), and purpose of use complete (information has all its sub-parts for performing a task at hand). Considering activity `T(8)`, `trader suggestions` has been provided to the `investor` through IP-Provision (it is value complete), and such information is not composite information, i.e., it has all of its parts (it is purpose of use complete) . In case, `trader suggestions` was not provided through Integrity Preserving (IP) provision, or it is composite and it misses any of its sub parts, *Pro13* will notify the designer that such information is incomplete for read.

Pro14 states that a WFA-net should not include any activity that reads information, and such information is inconsistent from the perspective of the actor who is responsible for the activity achievement. Information is consistent for read, if it has only one reader taking into consideration its purpose of use, or it has multiple readers for the same purpose of use, and all of them have the same read-time. Considering activity `T(8)`, the `investor` is the only reader of `trader suggestions`, i.e., such information is consistent for read. In case, there was another reader of such information with the same purpose of use, and they do not have the same read-time, *Pro14* will notify the designer that such information is inconsistent for read.

Pro15 states that a WFA-net should not include any activity that modifies information, and the actor who is responsible for achieving such activity does not have a modify permission. Considering activity `T(10)`, if the `investor` does not have a modify permission concerning `investor's orders`, *Pro15* will notify the designer that such information cannot be modified, since the responsible actor does not have modify permission.

Pro16 states that a WFA-net should not include any activity that sends information, and the actor who is responsible for its achievement does not have a send permission concerning such information. Considering activity `T(10)`, if the `investor` does not have a send permission concerning `investor's orders`, *Pro16* will notify the designer the responsible actor does not have a send permission.

Pro17 states that a WFA-net should not include any activity that sends information, and such information is inaccurate at its destination from the perspective of the actor (sender) who is responsible for the activity achievement. Information is accurate at its destination, if it has not been inappropriately modified during its transfer (trustworthiness of the provision). Considering activity `T(10)`, the `investor` sends `investor's orders` to `NASDAQ` through `Best trading`, if `Best trading` inappropriately modifies such information (modify without trust), *Pro17* will notify the designer that such information might be inaccurate at its destination.

Pro18 states that a WFA-net should not include any activity that sends information, and such information is incomplete at its destination from the perspective of the actor who is responsible for the activity achievement. Information is complete at its destination, if it has been provided through IP-Provision from its source to its destination, which guarantees that it has been preserved against lost and corruption during its transfer. Considering activity `T(10)`, if the `investor` sends `investor's orders` to `NASDAQ` through normal (P) provision, *Pro18* will notify the designer that such information is incomplete at its destination.

Pro19 states that a WFA-net should not include any activity that sends information, and such information is invalid at its destination from the perspective

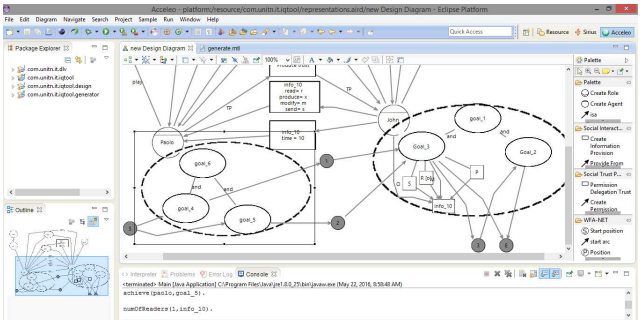

Fig. 6 Screen-shot of the prototype tool

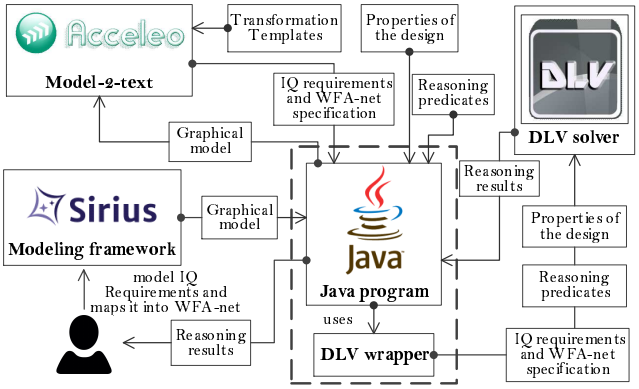

Fig. 7 Prototype tool architecture

of the actor who is responsible for the activity achievement. Information is valid at its destination, if its transfer (provision) time is less than its send time. Considering activity `T(10)`, the `investor` sends `investor's orders` to `NASDAQ` within 20 seconds (send time), if the provision time is more than 20 seconds, *Pro19* will notify the designer that such information is invalid at its destination.

Pro20 states that a WFA-net should not include any activity that has been prevented from being fired. Activities might be prevented from being fired due to several reasons. For example, the responsible actor does not have the capability to achieve the activity (goal), the responsible actor is not trusted for achieving the activity. Moreover, an activity might be prevented because of IQ related properties, e.g., activity is not able to produce, read, modify and/or send information, because the responsible actor does not have the required permission. Furthermore, an activity might be prevented because of reading inaccurate, incomplete, etc. information. Considering `T(4)`, the `trader` (responsible actor) is distrusted by the `investor` to achieve such activity (shown in Figure 3), i.e., such activity will be prevented from being achieved.

Pro21 states that the `End` position in a WFA-net should be reached, i.e., there should be at least one activity when fired the WFA-net reaches its `End` position. Consider `P(E)` in Figure 4 for example, both of `T(4)` or `T(11)` have outgoing arcs to `P(E)`, if none of them has fired, `P(E)` will not be reached and *Pro21* will notify the designer that the process has been terminated without reaching its end.

5 Implementation and evaluation

Evaluation aims to demonstrate the utility and efficacy of a design artifact. We evaluated our approach on a simulation basis following [35], i.e., developing a prototype tool and test its applicability with artificial data, a screen-shot of the tool is shown in Figure 6. Therefore, we developed a prototype implementation[8] to test the approach, specifically its ability to model and analyze IQ requirements in BPs. In what follows, we briefly describe the prototype architecture, and then we discuss its applicability over two scenarios abstracted from the stock market domain.

Prototype implementation. Our prototype has been developed depending on Eclipse Integrated Development Environment (IDE), and it consists of four main components (the tool architecture is depicted in Figure 7): *(1) Control component* (JAVA-based program), controls and coordinates the three other components; (2) A graphical user interface (GUI)[9]: supports designers during the design of the system-to-be, where the BP is executed, and then maps the requirements model into the WFA-net; (3) Model-to-text transformation: supports the translation of the graphical BP model into disjunctive Datalog formal specifications depending on Acceleo[10]; and (4) automated reasoning support (DLV system[11]) that takes the disjunctive Datalog specifications as an input, and then perform the required analysis to verify the correctness and completeness of the BP model against the properties of the design.

Applicability. We evaluated our approach by showing its utility and efficacy in modeling and analyzing IQ requirements in BPs along with its effectiveness in capturing any violation to the properties of the design by applying it to two scenarios abstracted from the Flash Crash case study[12].

In particular, we modeled each scenario as it might occur in the real world in its social and organizational context (requirements model), and then we modeled the process that represents the scenario by mapping the IQ requirements in terms of its goals into activi-

[8] The prototype tool is available at `goo.gl/Iy1BjR/`
[9] Developed by Sirius `https://goo.gl/b4Mwjt`
[10] https://goo.gl/vC6vvv
[11] http://www.dlvsystem.com/dlv/
[12] For more information about the case study refer to [26]

ties of the WFA-net taking into consideration the rules to define building blocks, consistency, sequencing and refinement constraints. Moreover, we translated the resulting model (IQ requirements and WFA-net models) into disjunctive Datalog specification, which enables us to run the automated analysis to test the analysis ability in discovering any violation to the properties of the design. In what follows, we discuss each of these scenarios:

Scenario 1. A `stock investor` aims to make profit from trading securities in `NASDAQ`. The process for achieving such goal can be achieved either by delegating the goal of trading securities to `Best trading`, or by achieving such goal on her own. In case the investor wants to achieve the goal by herself, she needs to decide the right trading orders either by depending on consultant suggestions or by depending on the trader's suggestions. Moreover, the investor needs to produce orders and send them to `NASDAQ` through a stock trader (`Best trading`). If the trade was successfully performed, the trade settlement, which results from performing such trade is sent back through `Best trading` to the investor to finalize the trade.

The process in terms of its activities is shown in Figure 4, where the process has been mapped from the requirements model into activities of the WFA-net taking into consideration the different consistency, sequencing and refinement constraints[13]. The process has two parallel paths from `S`tart to `E`nd position[14]. In the first path, it has only one activity `T4: Trade by a trader` that is connected directly with the `S`tart position, i.e., `T4` is enabled. If `T4` fires, the `E`nd position will be reached, i.e., the process has succeeded in reaching its end. However, `T4` will not fire, since the `investor` does not trust `Best trading` (responsible actor) for achieving `T4`. Without trust there is no guarantee that `Best trading` will achieve such activity.

The second path starts with three parallel activities, `T1: Perform trade`, `T2: Perform trade`, and `T3: Produce securities' assessments`. `T1` and `T2` have two outgoing arcs to positions `P1` and `P2` respectively. When both `T1` and `T2` fire, we have two tokens at `P1` and `P2` respectively, and `T5: Analysis overall trading environment` will be enabled. However, `T5` may not fire if there is any delay concerning `trade info-NYSE` and/or `trade info-NASDAQ` information, which makes such information invalid for read. For example, one main reason of the Flash Crash was the delay of NYSE quotes (`Trade info-NYSE`) provision to `CTA` [51, 60], which resulted in invalid `CQS-info` for many stock traders. Therefore, any analysis performed depending on such information was wrong, which led to wrong trading decisions on the traders' side [79, 66]. In case there is no delay concerning `trade info-NYSE` and `trade info-NASDAQ`, `T5` fires, the two tokens in `P1` and `P2` are removed, and a token is placed in `P5`, since there is an outgoing arc from `T5` to position `P5`.

On the other hand, `T3` is enabled, and when it fires two tokens are produced in `P3` and `P4` respectively. In this context, both of `T6: Analysis by itself` and `T7: Produce consultant suggestions` are enabled. However, `T6` will not fire since `securities' assessments` is inaccurate for read, i.e., there is no trust for producing such information between producer (`credit assessments firm`) and reader (`Best trading`). In the stock market domain, providing fraudulent/falsified (untrustworthy/inaccurate information) assessment (also trading suggestions) is, usually, due to conflict of interest. The Enron scandal [53] is an example, where the conflict of interest resulted in producing untrustworthy/inaccurate information. More specifically, Arthur Andersen was the audit company of Enron, and it provides fraudulent/falsified information concerning the assets of Enron [49], which led to an incorrect estimation of Enron's securities by stock traders.

If this violation was resolved, and `securities' assessments` was verified trustworthy (there is a trust for producing concerning such information), `T6` fires, the two tokens in `P3` and `P5` are removed, and a token is produced in `P6`. On the other hand, `T7` might face the same issue while reading `securities' assessments`. If this violation is resolved, and the `securities' assessments` is verified trustworthy, `T7` fires, the token in `P4` is removed, and a token is produced in `P7`. At this point, both of `T8: Analysis by trader suggestions` and `T9: Analysis by consultant suggestions` are enabled, since there are tokens in `P6` and `P7`. However, `T8`/`T9` may not fire, if the trustworthiness of `trader suggestions`/`consultant suggestions` is not verified.

If these two violations are resolved, `T8`/`T9` fire, the token at `P6`/`P7` is removed, and a token at `P8` is produced, which enables activity `T10: Produce trading orders`. `T10` may not fire, if there is no trust for send between `investor` and `Best trading`, who is responsible for sending the investor's orders to `NASDAQ`. Trust for send is another issue in the stock market domain, since traders may sell the orders they are responsible for performing by forwarding them to other trading companies for a fee ("payment for order flow"). For example,

[13] We mainly focus on the ability of the automated analysis in detecting any violation to the properties of the design

[14] All possible configurations concerning this process are shown in Figure 5

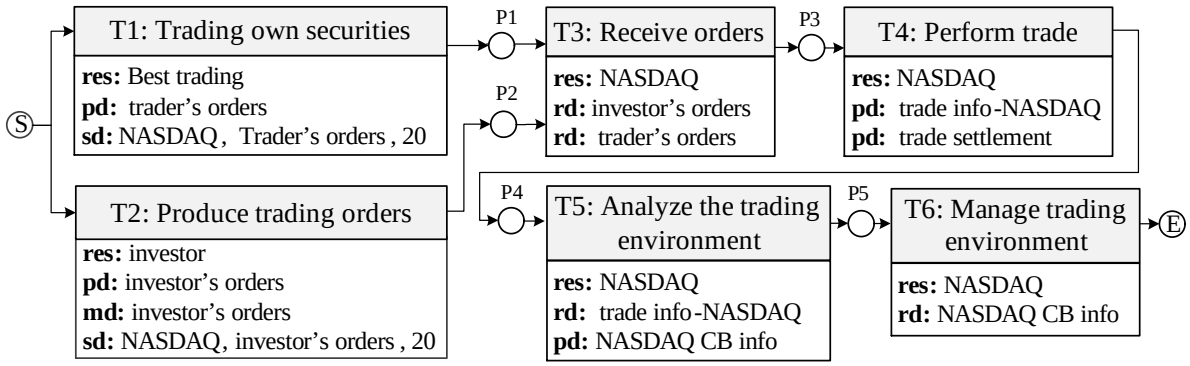

Fig. 8 A WFA-net concerning a stock market process for facilitate trading among stock traders

Citadel LLC (high-frequency trader) paid TD Ameritrade (trader) hundreds of millions of dollars each year to forward their orders to Citadel LLC [47]. When `T10` fires, the token at `P8` will be removed, and a token will be produced at `P9`, which enables activity `T11: Finalize trade`. When `T11` fires, the `End` position of the WFA-net will be reached, and we can say that the process has reached its end successfully.

Scenario 2. `NASDAQ` aims to make a profit by facilitating securities trading among stock traders. The process for achieving such goal starts by receiving orders from both investors and traders, match such orders, and perform trades when orders match. Moreover, `NASDAQ` should guarantee a stable and fair trading environment for its traders, which can be done by analyzing the trading environment, and slow or even stop the trading activities when required to prevent a market crash. The process is shown in Figure 8.

The process has only one path from `Start` to `End` position, where both of `T1: Trading own securities` and `T2: Produce trading orders` are enabled, since they are linked with the `Start` position. `T1` will not fire, since `trader's orders` will not be valid at its destination (send time is bigger than provision time). When this violation is fixed (e.g., relaxing the send time, using faster provision time), `T1` will fire, and a token will be produced at `P1`. We consider that `T2`[15] has fired, and a token is produced at `P2`.

`T3: Receive orders` is enabled, since there are two tokens in `P1` and `P2` respectively. However, `T3` will not fire since `trader's orders` is inaccurate for read, i.e., there is no trust for producing such information between producer (`Best trading`) and reader (`NASDAQ`). In particular, `Best trading` is playing an `HFT` role, where a `HFT` have the capability of manipulating the trading environment by providing falsified orders (e.g., flickering quotes[16]) in order to influence the prices of securities before starting its real trades. The suspicious behavior of some HFTs was considered as a main reason that led to the Flash Crash [66, 41]. In particular, if orders that have been provided by HFTs were not considered trustworthy for granted, the Flash Crash might have been avoided [30].

Note that if the trader was playing a `Market maker` role, it might provide orders with unbelievable prices (e.g., stub quotes[17]) to fulfill its obligations concerning the provision of sell/buy orders in the market. According to [41], during the Flash Crash, over 98% of all trades were executed at prices within 10% of their values before the crash because of the stub quotes. However, such issue can be avoided, if the market verifies the believability of any order it receives, i.e., applies a believability check while reading such orders.

After resolving the violation at `T3` (e.g., accepting only trustworthy information), `T3` fires, the tokens at `P1` and `P2` are removed, and a token at `P3` is produced. Having a token at position `P3` enables `T4: Perform trade`[18], and when it fires, the token at `P3` is removed, and a token at `P4` is produced, which enables `T5: Analyze the trading environment`. When activity `T5` fires, the token at `P4` is removed, and a token at `P5` is produced, which enables `T6: Manage trading environment`.

Activity `T6` will not fired, since `NASDAQ CB info` is incomplete for managing the trading environment. A main reason of the Flash Crash was inefficient coordination among the CBs of the trading markets [32], i.e., each market conducted analysis of its own trading environment, and based on such analysis, it could employ its

[15] `T2` is the same as activity `T10` in Scenario 1, and there is no need to discuss it again

[16] Orders that last very short time, which make them unavailable for most of the traders [46]

[17] Orders with prices far away from the current market prices [41]

[18] `T4` is the same as activity `T2` in Scenario 1

CB when needed. For instance, during the Flash Crash CME employed its CB, while NYSE did not [73].

As previously mentioned, for each security there is only one primary listing market, i.e., markets need to coordinate their CBs with the CB of the primary listing market to prevent a potential market crash. In this context, we can solve the coordination problem, by considering the CB information of the primary listing market (`CME CB info`) as a sub part (part of) of both `NYSE CB info` and `NASDAQ CB info`, which guarantee the completeness of such information for managing the trading environments of both `NYSE` and `NASDAQ`.

After considering `CME CB info` as a sub part of both of `NYSE CB info` and `NASDAQ CB info`, they become complete for managing the trading environment of `NYSE` and `NASDAQ` respectively. This raises another problem since `NYSE` and `NASDAQ` have became *inter-dependent readers* for `CME CB info`, i.e., `CME CB info` should be consistent between them. However, `CME CB info` is inconsistent between them, since they have two different *read times* concerning `CME CB info`, where the *read time* of *NASDAQ* is 130 ms, and the *read time* of `NYSE` is 146 ms [47] concerning `CME CB info`. This situation can be resolved by unifying the read time of `CME CB info` between `NYSE` and `NASDAQ`, which can be done by unifying the provision time of `CME CB info` to both of them. After resolving the violation at activity `T6`, it fires and the `End` position of WFA-net is reached, i.e., the process has succeeded in reaching its end.

6 Approach limitations and threats to validity

After presenting and discussing our approach, we discuss its limitations and threats to validity.

Approach limitations. We have identified the following limitations:

Binary requirement satisfaction: the approach only deals with binary requirement satisfaction (e.g., a goal can be either satisfied or denied), i.e., it does not support a qualitative requirements reasoning. Similarly, the approach only deals with binary IQ requirement satisfaction, i.e., information can be either accurate or inaccurate, believable or unbelievable, etc.

All IQ dimensions have the same priority: all IQ dimensions have the same importance to the system, i.e., all of them have the same priority. For example, information accuracy has the same importance for the system as information consistency, completeness, etc. However, in some domains one IQ dimension might be more important than others, which can be reflected by assigning different priorities to IQ dimensions based on their importance to the system.

Only one BP: the approach cannot deal with more than one BP at the same time, i.e., it is not possible to model more than one BP at the same time.

No customized analysis: the Tool does not support customized analysis, it only supports verifying all the properties of the design, i.e., a user cannot choose which properties of the design to be verified.

Tool installation: the installation of the Tool is not user-friendly, since it requires several applications to be installed on the host machine (e.g., Java, Sirius, Acceleo, etc.) to run appropriately.

Approach threats to validity. We discuss here the threats to the internal and external validity of our evaluation of the approach based on the Flash Crash case study.

Internal validity: is concerned with factors (third factors) that have not been considered in the study, and they could have influenced the investigated factors in the study [78, 61]. We have identified the following internal threats:

Other factors might lead to the Flash Crash: our analysis have identified several IQ related vulnerabilities that have led or contributed to the Flash Crash. However, other factors might be involved as well, which we were not able to identify in our analysis. Further analysis of the Flash Crash is required to verify that the factors we considered are enough (if tackled) to avoid such a crash, or identify unrevealed factors.

Experimenter bias: occurs when the experimenter influences the outcome of the study. For example, an experimenter might tend not to look for evidence that might negate its expectations. To avoid such threat, our role as experimenters was limited to modeling the scenarios as they might occur in the real world, and then running the automated analysis. Moreover, our findings concerning the Flash Crash case study have been reported by other researchers/experts in several fields.

External validity: is concerned with to what extent the results of the study can be generalized [61]. We have identified the following external threats:

No extensive evaluation: the approach has been applied to only one domain (stock market), which threatens the generalization of our findings. Probably, applying the approach in other domains might reveal undetected inadequacies for modeling and analyzing IQ requirements in BPs.

Completeness of the properties of the design: we did not evaluate the completeness of the proposed properties of the design with domain experts. However, we have identified these properties based on an extensive analysis of available reports and studies about the Flash Crash (e.g., [51, 79, 66]).

7 Related Work

Traditionally, BPs literature has focused on control-flow perspective of the processes with less emphasis on information perspective. However, some efforts have been devoted to the design of data-aware processes with main emphasis on information-flow, which enables for avoiding errors that result from using information that is not yet available in the BP. For instance, generic patterns of how data are addressed in BP have been presented in [62], and exception management mechanisms to deal with data unavailability in the BP has been introduced in [63]. While Trčka et al. [76] proposed data anti-patterns that represent undesirable data-flow behaviors in BPs. Sadiq et al. [64] introduced a method that identifies the requirements of data flow modeling in workflow specifications. Combi and Gambini [17] presented *A* modeling language for capturing control-flow along with data-flow relevant concerns. Soffer [71] introduced a theoretical approach that captures data inaccuracy along with the expected results of depending on such data in BPs.

Other approaches for integrating data flow with control flow of BPs include case handling [80], ad-hoc process modifications approaches [59], and artifact-centric [16]. In addition, Zhao et al. [87] propose an Artifact-Centric Business Process Model (ABPM) that allows representing artifacts (data objects), which includes data associated with a business object, data about its overall life cycle and relationships to other artifacts. While Calvanese et al. [12] introduce an artifact-centric approach that is able to characterize when one artifact-centric workflow *dominates* another one. In [43], the authors develop an approach for identifying business entities from activity-centric process models and then transforming such models into information-centric business process models.

Bhattacharya et al. [6] develop an approach for workflow design that is founded on a data-centric perspective. Their approach focuses primarily on business artifacts and how they can be used to provide core elements of an overall design methodology for business operations. Sun et al. [75] propose the SeGA framework that supports the separation of data and BP execution. Moreover, they introduce the concept of a self-guided artifact that extends artifact-centric BP models by capturing all needed data for a BP throughout its execution. Finally, Sun et al. [74] develop an approach for modeling data for business processes, which represents data used by a process as a hierarchically structured business entity characterized by keys, local keys, update constraints, and a set of data mapping rules defining exact correspondence between entity data values and values in the enterprise database.

Deutsch et al. [20] propose *TNest* that is a data-centric workflow modeling language, which allows for expressing data dependencies along with time constraints. While Sidorova et al. [68] proposed a new data-aware soundness notion, WFD-net that is able to address data-flow issues along with the control-flow of BPs. Furthermore, Guerra-García et al. [33] introduced a Model Driven Architecture (MDA) for the management of Data Quality (DQ) during the design and development of Web applications. In particular, they propose a meta-model and a UML profile for capturing and specifying DQ requirements (DQ_WebRE). Finally, Cappiello et al. [13] proposed a methodology to support BP designers in identifying DQ requirements, and selecting the required actions to satisfy such requirements during the design of BPs.

Outside the BP area, several approaches for improving IQ by design have been proposed. For instance, Wang [83] proposed the Total Data Quality Management (TDQM) methodology for delivering high-quality information products (IP) to information consumers. They introduce the concept of Information Product (IP) to emphasize the fact that the information output from an information manufacturing system has a value that is transferable to the consumer. Furthermore, Ballou et al. [4] presented Information Manufacturing System (IMS) that considers four attributes of IP, namely: timeliness, quality, cost, and value of information products. Shankaranarayanan et al. [67] extended the work of Ballou, and proposed a formal modeling method for creating an IP-MAP. Scannapieco et al. [65] relied on the IP-MAP to propose IP-UML that is an engineering approach developed to improve data quality in a single organization. However, all the previously mentioned approaches were not designed to capture the IQ requirements in their social and organizational context.

On the other hand, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) published several quality models including ISO/IEC 9126 [37] and ISO/IEC 25010 [39]. However, the main focus of these standards is software quality not data/information quality. Natale et al. [52] tried to solve this problem by linking software quality attributes proposed in ISO/IEC 9126 into a set

of data quality dimensions proposed by Thomas Redman [58]. ISO/IEC 25012 [38] is a standard for data quality, which analyzes data quality in terms of fifteen quality attributes (dimensions). ISO/IEC 25012 can be used as a baseline for our model since our model considers most of the IQ dimensions proposed by ISO/IEC 25012. Yet we adopted our previous work since we do not need only the IQ dimensions but also how such dimensions can be analyzed in their social and organizational context, which is not supported by ISO/IEC 25012.

Finally, combining goal models and BPs is not new, for example, Cysneiros and Yu [18] discuss agents autonomy in modeling and supporting business processes. While Koliadis et al. [42] proposed a preliminary work for mapping *i** to BPMN. Lapouchnian et al. [44] propose a requirements-driven approach for BP design that uses requirements goal models to capture alternatives in process configuration. However, to the best of our knowledge, there is no previous work in Goal-Oriented Requirements Engineering that models IQ requirements and maps such requirements into a BP.

8 Conclusions and Future Work

In this paper, we discussed the importance of modeling and analyzing IQ requirements during the early phases of the BPs design, and we advocate that such requirements should be analyzed in their social and organizational context. In particular, we proposed an approach for modeling and analyzing IQ requirements in BPs from a socio-technical perspective. More specifically, the approach relies on our goal-oriented framework [30, 28] for modeling and analyzing IQ requirements in their social and organizational context, and then it proposes several types of constraints to guarantee the correct and proper mapping of such requirements into workflow net with actors (WFA-net) that is a BP modeling language we proposed. Moreover, we provided a detailed execution semantics for the WFA-nets. In addition, we discussed the analysis techniques our approach proposes, which support the verification of the control-flow, information-flow, and IQ requirements of BPs. We evaluated our approach by developing a prototype tool and test its applicability by modeling and analyzing the IQ requirements of two realistic scenarios abstracted from the Flash Crash.

For future work, we intend to extend the IQ dimensions we considered in this work, and we aim to better investigate inter-dependencies among IQ dimensions. We intend to provide a more expressive analysis for IQ related aspects rather than the binary one, which use only two values to evaluate IQ related concepts (e.g., accurate or inaccurate, believable or unbelievable, etc.). Another research thread under investigation is prioritizing IQ dimensions based on their importance for achieving a task at hand. Enhancing the modeling component by adopting the multi-view modeling is also on our list for future work.

On the other hand, we are investigating how to extend the approach to deal with more than one BP at the same time. Moreover, we intend to provide a customized analysis concerning the properties of the design. We aim to better validate our approach by applying it to other complex case studies that belong to different domains. Finally, we are planning to perform a set of experiments with designers to evaluate the adequacy of our proposed approach for modeling IQ requirements, mapping such requirements into WFA-net, and using the automated analysis to verify the WFA-net model.

Acknowledgements This research has received funding from the ERC advanced grant No. 267856, "Lucretius: Foundations for Software Evolution", `http://www.lucretius.eu/`, and the European Union's Horizon 2020 research and innovation programme under grant agreement No. 653642 - VisiOn.

References

1. Aguilar-Saven, R. S. Business process modelling: Review and framework. *International Journal of production economics 90*, 2 (2004), 129–149.
2. Aldridge, I. *High-frequency trading: a practical guide to algorithmic strategies and trading systems*. John Wiley & Sons, 2013.
3. Ballou, D., and Pazer, H. Modeling data and process quality in multi-input, multi-output information systems. *Management science 31*, 2 (1985), 150–162.
4. Ballou, D., Wang, R., Pazer, H., and Tayi, G. K. Modeling information manufacturing systems to determine information product quality. *Management Science 44*, 4 (1998), 462–484.
5. Batini, C., and Scannapieco, M. *Data quality: concepts, methodologies and techniques*. Springer-Verlag, 2006.
6. Bhattacharya, K., Hull, R., Su, J., et al. A data-centric design methodology for business processes. *Handbook of Research on Business Process Modeling* (2009), 503–531.
7. Biersack, E. W. Performance evaluation of forward error correction in ATM networks. In *ACM SIGCOMM Computer Communication Review* (1992), vol. 22, ACM, pp. 248–257.

8. Bihlmeyer, R., Faber, W., Ielpa, G., Lio, V., and Pfeifer, G. DLV-user manual. *The DLV Project* (2009).
9. Bovee, M., Srivastava, R. P., and Mak, B. A conceptual framework and belief-function approach to assessing overall information quality. *International Journal of Intelligent Systems 18*, 1 (2003), 51–74.
10. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems 8*, 3 (2004), 203–236.
11. Buneman, P., Khanna, S., and Wang-Chiew, T. Why and where: A characterization of data provenance. In *International conference on database theory* (2001), Springer, pp. 316–330.
12. Calvanese, D., De Giacomo, G., Hull, R., and Su, J. Artifact-centric workflow dominance. In *Service-Oriented Computing*. Springer, 2009, pp. 130–143.
13. Cappiello, C., Caro, A., Rodriguez, A., and Caballero, I. An approach to design business processes addressing data quality issues. In *ECIS* (2013), p. 216.
14. Cliff, D. The Flash Crash of May 6th 2010: WTF. Tech. rep., Mimeo, University of Bristol, 2011.
15. Cohen, F. A cryptographic checksum for integrity protection. *Computers & Security 6*, 6 (1987), 505–510.
16. Cohn, D., and Hull, R. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull 32*, 3 (2009), 3–9.
17. Combi, C., and Gambini, M. Flaws in the flow: The weakness of unstructured business process modeling languages dealing with data. In *On the Move to Meaningful Internet Systems: OTM 2009*. Springer, 2009, pp. 42–59.
18. Cysneiros, L. M., and Yu, E. Addressing agent autonomy in business process management-with case studies on the patient discharge process. In *Proc. 2004 IRMA Conference* (2004).
19. Dai, C., Lin, D., Bertino, E., and Kantarcioglu, M. An approach to evaluate data trustworthiness based on data provenance. In *Workshop on Secure Data Management* (2008), Springer, pp. 82–98.
20. Deutsch, A., Hull, R., Patrizi, F., and Vianu, V. Automatic verification of data-centric business processes. In *Proceedings of the 12th International Conference on Database Theory* (2009), ACM, pp. 252–267.
21. Emery, F., and Trist, E. Socio-technical systems. management sciences, models and techniques. churchman cw et al, 1960.
22. Eric, S., and Mylopoulos, J. From ER to AR - modelling strategic actor relationships for business process reengineering. In *International Conference on Conceptual Modeling* (1994), Springer, pp. 548–565.
23. Eriksson, H.-E., and Penker, M. Business modeling with UML. *Business Patterns at Work, John Wiley & Sons, New York, USA* (2000).
24. Fisher, C. W., and Kingma, B. R. Criticality of data quality as exemplified in two disasters. *Information & Management 39*, 2 (2001), 109–116.
25. Gharib, M., and Giorgini, P. Modeling and analyzing information integrity in safety critical systems. In *Advanced Information Systems Engineering Workshops* (2013), Springer, pp. 524–529.
26. Gharib, M., and Giorgini, P. Detecting conflicts in information quality requirements: the May 6, 2010 Flash Crash. Tech. rep., Universitá degli studi di Trento, 2014.
27. Gharib, M., and Giorgini, P. Analyzing trust requirements in socio-technical systems: A belief-based approach. In *IFIP Working Conference on The Practice of Enterprise Modeling* (2015), Springer, pp. 254–270.
28. Gharib, M., and Giorgini, P. Dealing with information quality requirements. In *International Conference on Enterprise, Business-Process and Information Systems Modeling* (2015), Springer, pp. 379–394.
29. Gharib, M., and Giorgini, P. A goal-based approach for automated specification of information quality policies. In *9th International Conference on Research Challenges in Information Science (RCIS)* (2015), IEEE, pp. 139–150.
30. Gharib, M., and Giorgini, P. Modeling and reasoning about information quality requirements. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (2015), Springer, pp. 49–64.
31. Gharib, M., and Giorgini, P. Modeling and reasoning about information quality requirements in business processes. In *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2015, pp. 231–245.
32. Gomber, P., Haferkorn, M., Lutat, M., and Zimmermann, K. The effect of single-stock circuit breakers on the quality of fragmented markets. In *FinanceCom* (2012), pp. 71–87.
33. Guerra-García, C., Caballero, I., and Piattini, M. Capturing data quality requirements for

web applications by means of dq_webre. *Information Systems Frontiers 15*, 3 (2013), 433–445.
34. Hamming, R. Error detecting and error correcting codes. *Bell System technical journal 29*, 2 (1950), 147–160.
35. Hevner, A. R., March, S. T., Park, J., and Ram, S. Design science in information systems research. *MIS quarterly 28*, 1 (2004), 75–105.
36. Holden, C. W., and Jacobsen, S. The breakdown of standard microstructure techniques: And what to do about it. *Available at SSRN 1911491* (2011).
37. International Organization for Standardization/International Electrotechnical Commission. *ISO/IEC 9126: Information Technology-Software Product Evaluation-Quality Characteristics and Guidelines for Their Use*. ISO/IEC, 1991.
38. International Organization for Standardization/International Electrotechnical Commission. ISO/IEC 25012: Software engineering-software product quality requirements and evaluation (square)-data quality model. *ISO/IEC* (2009).
39. International Organization for Standardization/International Electrotechnical Commission. *ISO/IEC 25010:2011: Systems and software engineering – Systems and software product Quality Requirements and Evaluation (SQuaRE)*. 2011.
40. Juran, J., Gryna, F., and Bingham, R. *Quality control handbook*, vol. 9. McGraw-Hill Book Company, Chapters, 1979.
41. Kirilenko, A. A., Kyle, A. S., Samadi, M., and Tuzun, T. The Flash Crash: The impact of high frequency trading on an electronic market. *Available at SSRN 1686004* (2015).
42. Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., and Ghose, A. Combined approach for supporting the business process model lifecycle. In *PACIS* (2006), Citeseer, p. 76.
43. Kumaran, S., Liu, R., and Wu, F. Y. On the duality of information-centric and activity-centric models of business processes. In *International Conference on Advanced Information Systems Engineering* (2008), Springer, pp. 32–47.
44. Lapouchnian, A., Yu, Y., and Mylopoulos, J. Requirements-driven design and configuration management of business processes. In *Business Process Management*. Springer, 2007, pp. 246–261.
45. Liu, L., and Chi, L. Evolutional data quality: A theory-specific view. In *IQ* (2002), pp. 292–304.
46. McInish, T., and Upson, J. Strategic liquidity supply in a market with fast and slow traders. *Available at SSRN 1924991* (2012).
47. Michael, L. Flash boys: a wall street revolt, 2014.
48. Mishkin, F. S. Policy remedies for conflicts of interest in the financial system. In *Macroeconomics, Monetary Policy and Financial Stability Conference* (2003), Citeseer.
49. Moore, D. A., Tetlock, P. E., Tanlu, L., and Bazerman, M. H. Conflicts of interest and the case of auditor independence: Moral seduction and strategic issue cycling. *Academy of Management Review 31*, 1 (2006), 10–29.
50. Murata, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE 77*, 4 (1989), 541–580.
51. Nanex, LLC. Nanex Flash Crash summary report. `http://www.nanex.net/FlashCrashFinal/FlashCrashSummary.html`, 2010. Accessed: 2014-05-30.
52. Natale, D., Scannapieco, M., Angeletti, P., Catarci, T., and Raiss, G. Qualità dei dati e standard ISO/IEC 9126: Analisi critica ed esperienze nella pubblica amministrazione italiana. In *Proceedings of the National Workshop on Sistemi in Rete nella Pubblica Amministrazione (in Italian), Roma, Italy* (2001).
53. Petrick, J. A., and Scherer, R. F. The Enron scandal and the neglect of management integrity capacity. *American Journal of Business 18*, 1 (2003), 37–50.
54. Pipino, L. L., Lee, Y. W., and Wang, R. Y. Data quality assessment. *Communications of the ACM 45*, 4 (2002), 211–218.
55. Prabhakaran, V., Arpaci-Dusseau, A., and Arpaci-Dusseau, R. Analysis and evolution of journaling file systems. In *Proceedings of the Annual USENIX Technical Conference* (2005), pp. 105–120.
56. Quinlan, S., and Dorward, S. Venti: a new approach to archival storage. In *Proceedings of the FAST 2002 Conference on File and Storage Technologies* (2002), vol. 4.
57. Redman, T. C. Improve data quality for competitive advantage. *Quality Control and Applied Statistics 41* (1996), 49–52.
58. Redman, T. C., and Blanton, A. *Data quality for the information age*. Artech House, Inc., 1997.
59. Reichert, M., Rinderle-Ma, S., and Dadam, P. Flexibility in process-aware information systems. *Transactions on Petri Nets and Other Models of Concurrency II* (2009), 115–135.
60. Rose, C. The Flash Crash of May 2010: Accident or market manipulation? *Journal of Business &*

Economics Research 9, 1 (2011), 85.
61. Runeson, P., and Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering 14*, 2 (2009), 131–164.
62. Russell, N., ter Hofstede, A., Edmond, D., and van der Aalst, W. Workflow data patterns: Identification, representation and tool support. *Conceptual Modeling* (2005), 353–368.
63. Russell, N., van der Aalst, W., and ter Hofstede, A. Exception handling patterns in process-aware information systems. *BPM Center Report BPM-06-04, BPMcenter.org* (2006), 06–04.
64. Sadiq, S., Orlowska, M., Sadiq, W., and Foulger, C. Data flow and validation in workflow modelling. In *Proceedings of the 15th Australasian database conference-Volume 27* (2004), Australian Computer Society, Inc., pp. 207–214.
65. Scannapieco, M., Pernici, B., and Pierce, E. IP-UML: Towards a methodology for quality improvement based on the IP-MAP framework. In *IQ* (2002), pp. 279–291.
66. Securities and Exchange Commission and others. Findings regarding the market events of May 6, 2010. *Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues* (2010).
67. Shankaranarayanan, G., Wang, R. Y., and Ziad, M. IP-MAP: Representing the manufacture of an information product. In *IQ* (2000), pp. 1–16.
68. Sidorova, N., Stahl, C., and Trčka, N. Workflow soundness revisited: checking correctness in the presence of data while staying conceptual. In *Advanced Information Systems Engineering* (2010), Springer, pp. 530–544.
69. Simmhan, Y. L., Plale, B., and Gannon, D. A survey of data provenance in e-science. *ACM Sigmod Record 34*, 3 (2005), 31–36.
70. Sobolewski, J. S. Cyclic redundancy check. In *Encyclopedia of Computer Science*. John Wiley and Sons Ltd., Chichester, UK, 2003, pp. 476–479.
71. Soffer, P. Mirror, mirror on the wall, can I count on you at all? exploring data inaccuracy in business processes. *Enterprise, Business-Process and Information Systems Modeling* (2010), 14–25.
72. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J., and Paige, R. Large-scale complex IT systems. *Communications of the ACM 55*, 7 (2012), 71–77.
73. Subrahmanyam, A. Algorithmic trading, the Flash Crash, and coordinated circuit breakers. *Borsa Istanbul Review 13*, 3 (2013), 4–9.
74. Sun, Y., Su, J., Wu, B., and Yang, J. Modeling data for business processes. In *2014 IEEE 30th International Conference on Data Engineering* (2014), IEEE, pp. 1048–1059.
75. Sun, Y., Su, J., and Yang, J. Separating execution and data management: A key to business-process-as-a-service (BPaaS). In *International Conference on Business Process Management* (2014), Springer, pp. 374–382.
76. Trčka, N., van der Aalst, W., and Sidorova, N. Analyzing control-flow and data-flow in workflow processes in a unified way. Tech. Rep. 08-31, Eindhoven University of Technology, 2008.
77. Trčka, N., van der Aalst, W., and Sidorova, N. Data-flow anti-patterns: Discovering data-flow errors in workflows. In *Advanced Information Systems Engineering* (2009), Springer, pp. 425–439.
78. Trochim, W., and Donnelly, J. *The Research Methods Knowledge Base*. Cengage Learning, 2006.
79. U.S. Commodity Futures Trading Commission. Preliminary findings regarding the market events of May 6, 2010. *Report of the Staffs of the CFTC and SEC to the Joint Advisory Committee on Emerging Regulatory Issues* (2010).
80. Van der Aalst, W., Weske, M., and Grunbauer, D. Case handling: a new paradigm for business process support. *Data & Knowledge Engineering 53*, 2 (2005), 129–162.
81. van der Aalst, W. M. The application of petri nets to workflow management. *Journal of circuits, systems, and computers 8*, 01 (1998), 21–66.
82. Wand, Y., and Wang, R. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM 39*, 11 (1996), 86–95.
83. Wang, R. A product perspective on total data quality management. *Communications of the ACM 41*, 2 (1998), 58–65.
84. Wang, R., and Strong, D. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems* (1996), 5–33.
85. Yu, E. S.-K. *Modelling strategic relationships for process reengineering*. PhD thesis, University of Toronto, 1995.
86. Zannone, N. *A requirements engineering methodology for trust, security, and privacy*. PhD thesis, University of Trento, 2006.
87. Zhao, X., Su, J., Yang, H., and Qiu, Z. Enforcing constraints on life cycles of business artifacts. In *Theoretical Aspects of Software Engineering, 2009. TASE 2009. Third IEEE International Symposium on* (2009), IEEE, pp. 111–118.